2015

Vehicle Tracking and Classification via 3D Geometries for Intelligent Transportation Systems

William McDowell
University of Central Florida

VEHICLE TRACKING AND CLASSIFICATION VIA 3D GEOMETRIES FOR INTELLIGENT TRANSPORTATION SYSTEMS

by

WILLIAM KENNETH MCDOWELL
B.S. University of Central Florida, 2000
Master University of Florida, 2006

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical Engineering and Computer Science
In the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2015

Major Professor: Wasfy B Mikhael

ABSTRACT

In this dissertation, we present generalized techniques which allow for the tracking and classification of vehicles by tracking various Point(s) of Interest (PoI) on a vehicle. Tracking the various PoI allows for the composition of those points into 3D geometries which are unique to a given vehicle type. We demonstrate this technique using passive, simulated image based sensor measurements and three separate inertial track formulations. We demonstrate the capability to classify the 3D geometries in multiple transform domains (PCA & LDA) using Minimum Euclidean Distance, Maximum Likelihood and Artificial Neural Networks. Additionally, we demonstrate the ability to fuse separate classifiers from multiple domains via Bayesian Networks to achieve ensemble classification.

I dedicate this dissertation to my wife Jennifer and my children Emily, Bailey and Liam for their limitless support of my seemingly endless academic shenanigans. I would also like to thank my parents for their love and support and for teaching me the value of hard work and perseverance. It's through education our lives find purpose, but it's our loved ones which give it meaning.

ACKNOWLEDGMENTS

I would like to acknowledge the contributions of Dr. Wasfy Mikhael. One could not ask for a better advocate, teacher or friend. Additionally, I would like to thank Dr. Abhijit Mahalanobis for his guidance in balancing the challenges of academic publishing while working in industry.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS / ABBREVIATIONS

CW – Continuous Wave

ECEF – Earth Centered Earth Fixed

FMCW – Frequency Modulated Continuous Wave

GHT – Generalized Hough Transform

HOG – Histogram of Oriented Gradients

IFOV – Instantaneous Field of View

ILD – Inductive Loop Detector(s)

ITS – Intelligent Traffic Systems

KLT – Karhunen-Loeve Transform

LDA – Linear Discriminant Analysis

LOS – Line of Sight

MoG – Mixture of Gaussians

PCA – Principal Component Analysis

PDF – Probability Density Function

PoI – Point(s) of Interest

SIFT – Scale Invariant Feature Transform

SVM – Support Vector Machine

UAV - Unmanned Aerial Vehicle

WIM – Weigh-In-Motion

CHAPTER ONE:
INTRODUCTION

The demands on the surface transportation systems in the world are predicted to grow tremendously as the number of vehicles on the roads increases. In order to meet the challenges presented by the increased demand, Intelligent Transportation Systems (ITS) (also referred to as Intelligent Traffic Systems) require reliable traffic information to increase efficiency on existing roads and to determine priorities for new road construction [1] [2]. Additionally, many rural areas possess no existing infrastructure for ITS thus leaving gaps in the desired traffic information flow [3]. Given that the majority of ITS data gathering solutions are fixed installations, there exists a need for mobile platforms to collect reliable traffic parameter data. In order to supply the necessary planning information, various techniques have been developed to calculate macroscopic traffic flow parameters such as the volume, occupancy, and speed of traffic [4]. However, as the capabilities of ITS become increasingly complex and the areas it must service grow, the techniques which supply the vital traffic information must also evolve and improve.

Overview of Dissertation

In this Dissertation, we present a mobile generalized technique which allows for the tracking and classification of vehicles by tracking Point(s) of Interest (PoI) on a vehicle. In the context of this discussion, PoI are defined as physical locations on the vehicle which can be consistently located from image frame to image frame using image processing techniques. Tracking the PoI allows for the composition of those points into 3D geometries which are unique to a given vehicle type. We demonstrate this technique using passive, simulated image

based detector measurements and an inertial track formulation. We demonstrate the capability to classify the 3D geometries in multiple linear transform domains using Minimum Euclidean Distance, Maximum Likelihood and Artificial Neural Networks. Additionally, we demonstrate the ability to fuse separate classifiers from multiple linear transform domains via Bayesian Networks to achieve ensemble classification.

The requirements for this approach include the capability to consistently track the PoI on a vehicle, knowledge of the kinematics of the detector providing the measurements, and the ability to transform all measurements into a common inertial reference frame. The technique is independent of the image processing used to detect the PoI.

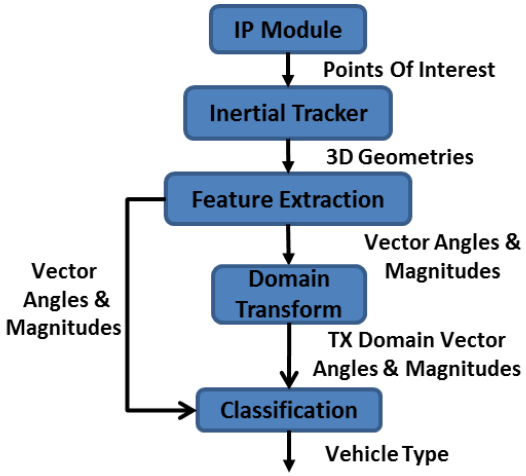

Fig. 1. System Diagram

The diagram in Figure 1 depicts the steps required for the proposed approach. Step one is to process a frame of video using an appropriate Image Processing (IP) Module capable of identifying the required PoI. For the context of this discussion, the frame of video can be defined as a frame of data from any type of passive imaging device, including

but not limited to, visible, multi-spectral, or hyperspectral data. For the purposes of our simulation, the IP Module transforms simulated 3D truth positions to noisy 2D angle measurements for use by the Inertial Tracker. In step two, the Inertial Tracker uses the angle measurements from each PoI to converge upon a 3D position vector representing the 3D location of each PoI in the reference frame. The 3D positions of the PoI are then grouped into a 3D geometry corresponding to the vehicle. In step three, the Feature Extraction module utilizes the 3D Geometry and extracts vector angles and magnitudes to be used as features for classification. Depending upon the classification approach being evaluated, the vector angles and magnitudes are either sent directly to the Classification module in their Native Spatial Domain or are pre-processed by the Domain Transform module. If the vector angles and magnitudes are sent to the Domain Transform module, they are transformed using either PCA or LDA before being sent to the Classification module. In the Classification Module, vehicle classification is performed with Minimum Euclidean Distance, Maximum Likelihood or Artificial Neural Networks using the vector angle and magnitude features represented in one or more domains.

Contributions

All of the approaches listed below are independent of the particular method being used to detect the PoI assuming the PoI consistently correlates to the same physical location on the vehicle. Additionally, the detection method must be able to consistently identify two or three PoI depending upon whether the two or three PoI approach is chosen. The unique contributions presented in this Dissertation are as follows:

1. A generalized, mobile, extensible, multi-feature, multiple PoI Kalman Filter 3D Geometry based framework for vehicle tracking and classification.
2. A two feature, two PoI, two Kalman Filter 3D Geometry based approach for single platform inertial tracking and classification of constant velocity vehicles in the Native Spatial, Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) domains using Minimum Euclidean Distance, Maximum Likelihood, Artificial Neural Networks and Bayesian Network based ensemble classifiers.
3. A three feature, two PoI, two Kalman Filter 3D Geometry based approach for single platform inertial tracking and classification of constant velocity vehicles in the Native Spatial, PCA and LDA domains using Minimum Euclidean Distance, Maximum Likelihood, Artificial Neural Networks and Bayesian Network based ensemble classifiers.
4. A seven feature, three PoI, three Kalman Filter 3D Geometry based approach for single platform inertial tracking and classification of constant velocity vehicles in the Native Spatial, PCA and LDA domains using Minimum Euclidean Distance, Maximum Likelihood, Artificial Neural Networks and Bayesian Network based ensemble classifiers.
5. A sixteen feature, three PoI, three Kalman Filter 3D Geometry based approach for single platform inertial tracking and classification of constant velocity vehicles in the Native Spatial, PCA and LDA domains using Minimum Euclidean Distance, Maximum Likelihood, Artificial Neural Networks and Bayesian Network based ensemble classifiers.

6. A two feature, two PoI, single Kalman filter constrained 3D Geometry based approach for single platform inertial tracking and classification of constant velocity vehicles in the Native Spatial, PCA and LDA domains using Minimum Euclidean Distance, Maximum Likelihood, Artificial Neural Networks and Bayesian Network based ensemble classifiers.
7. A three feature, two PoI, single Kalman filter constrained 3D Geometry based approach for single platform inertial tracking and classification of constant velocity vehicles in the Native Spatial, PCA and LDA domains using Minimum Euclidean Distance, Maximum Likelihood, Artificial Neural Networks and Bayesian Network based ensemble classifiers.
8. A seven feature, three PoI, single Kalman filter constrained 3D Geometry based approach for single platform inertial tracking and classification of constant velocity vehicles in the Native Spatial, PCA and LDA domains using Minimum Euclidean Distance, Maximum Likelihood, Artificial Neural Networks and Bayesian Network based ensemble classifiers.
9. A sixteen feature, three PoI, single Kalman filter constrained 3D Geometry based approach for single platform inertial tracking and classification of constant velocity vehicles in the Native Spatial, PCA and LDA domains using Minimum Euclidean Distance, Maximum Likelihood, Artificial Neural Networks and Bayesian Network based ensemble classifiers.

Approach

Using physics based principles and high fidelity simulation we demonstrate an approach to consistently track and identify a given vehicle type using the 3D geometry of the PoI. Angles and vector lengths are extracted from the 3D geometry and are used as features for classifying three separate vehicle types. The vehicle types chosen for the simulation were a Toyota Sienna Minivan, a Toyota Camry Sedan, and a Toyota Tundra pickup truck. The three vehicles were modeled in the simulation using their unique physical dimensions.

CHAPTER TWO:
LITERATURE REVIEW

The relevant literature and prior work can be grouped into the following sections: ITS Vehicle Detection & Classification Prior Work, Feature Calculation and Pattern Recognition. The ITS Vehicle Detection & Classification Prior Work section will provide a survey of existing technology and approaches. The Feature Calculation and Pattern Recognition sections will provide the background information necessary to understand the techniques proposed in this Dissertation.

ITS Vehicle Detection & Classification Prior Work

The design of ITS vehicle detection and classification approaches varies widely depending upon the nature of the sensors used. The sensors can be classified as being either intrusive versus non-intrusive and active versus passive. Intrusive sensors require that they are embedded in the road way in order to operate and consequently require changes to the road for implementation. Non-Intrusive sensors are able to operate with a degree of standoff and can be positioned next to or above the road way. Active sensors emit a signal of some kind which enable detection of the vehicles, whereas passive sensors operate on energy already existing in the environment. The most prevalent approaches are summarized in Table 1 [1].

Table 1
ITS Vehicle Detection & Classification Approaches

Technology	Passive or Active	Intrusiveness
Pneumatic Road Tubes	Passive	Intrusive
Inductive Loop Detectors	Passive	Intrusive
Magnetic Sensors	Passive	Intrusive
Weigh In Motion	Passive	Intrusive
Radar	Active	Non-Intrusive
Ultrasonic	Active	Non-Intrusive
Passive Acoustic	Passive	Non-Intrusive
Multi-Modal	Both	Both
Image Based	Passive	Non-Intrusive

In the following sections, we will provide an overview of the approaches listed in Table 1. The various approaches all measure one, two or three of the following macroscopic traffic flow parameters: volume, occupancy, and speed. Volume is a flow rate which measures the number of vehicles passing a point for a given period of time. Speed measures the rate at which the traffic is moving and occupancy is the percent of time the detector zone is occupied by a vehicle [4]. The techniques proposed in this dissertation are capable of measuring all three macroscopic parameters.

Pneumatic Road Tubes

Pneumatic road tubes detect vehicles by sending a burst of air pressure through a tube when a vehicle passes over the tube. The burst of air closes a switch, which in turn produces an electric current which can be monitored by a counter. The advantages of pneumatic road tubes include quick installation, low power usage, and low cost. Disadvantages include less than desired accuracy, durability, and not being suitable for high speed roads [1][5]. In [2] the authors investigate the accuracy of pneumatic road tubes and

finds that although the average error count for long term use tends to be near zero, shorter term use had an average error count close to ten percent. This leads to the conclusion that the level of inaccuracy is being masked by positive and negative counting errors. Additionally, the accuracy of speed data was found to be a concern.

Inductive Loop Detectors

An Inductive Loop Detector (ILD) consists of a loop of insulated wire buried in a shallow saw-cut in the roadway which is connected to an electronics unit. The electronics unit excites the wire with an electric current with a frequency between 10 KHz and 50 KHz creating an inductive loop. When a vehicle passes over or stops above the loop, the inductance is decreased; the oscillation frequency in the circuit increases and the electronics unit sends a signal to a controller allowing the vehicle to be counted. Inductive loop detectors are typically found in Single Loop and Dual Loop configurations. The data supplied by conventional Single Loop detectors includes vehicle passage, presence, count and occupancy. Dual Loop detectors, which are typically implemented as two consecutive single loops, provide the additional output of vehicle speed. The advantages of inductive loop detectors include low cost and maturity of the technology. Disadvantages include disruption of traffic for installation and repair and stress associated with temperature and traffic volume [1] [5]. Ongoing research continues to experiment with varying configurations of features and classifiers for Single Loop [6][7][8][9] Dual Loop [10][11] and inner/outer loop type detectors [12][13].

Magnetic Sensors

Magnetic sensors detect the presence of a ferrous metallic object, such as a vehicle, by monitoring for the changes in the Earth's magnetic field caused by the object. There are two types of magnetic sensors used for detecting vehicles, the two axis flux gate magnetometer and an induction coil magnetometer. The two axis flux gate magnetometer detects anomalies in the Earth's magnetic field caused by the presence of a vehicle. An induction coil magnetometer detects changes in the magnetic flux caused by the changing magnetic field values. Magnetic sensors are less susceptible than Inductive Loop Detectors to stress caused by traffic. Disadvantages include the need to modify roadways for installation or maintenance which results in road closures. Additionally, most magnetic sensors cannot detect stationary vehicles [1][5]. Areas of active research include development of new types of magnetic sensors [14] [15] as well as attempts to implement magnetic sensors as wireless sensor networks [16][17][18].

Weigh-In-Motion

There are three primary types of Weigh-In-Motion (WIM) sensors: Bending Plate, Piezoelectric and Capacitance Matts. Weigh-In-Motion systems estimate the vehicle weight as well as how much weight is carried by each wheel assembly[1]. WIM systems were first introduced in Alberta Canada in 1982 [19] and play a critical role in monitoring and controlling road damage due to overweight vehicles[20][21]. A Bending plate consists of a plate with strain gauges attached to one side. The gauges record the strain to the plate applied by the passing vehicles which can then be translated into weight. In practice Bending Plate WIM systems are typically coupled with two ILDs to measure vehicle speed [1].

Piezoelectric sensors generate an electric current in response to the pressure generated by a passing vehicle. Piezoelectric sensors are often coupled with two ILDs which allow the added capability of measuring vehicle speed [1]. Capacitance mats consist of two metal sheets with a dielectric material between them. The spacing of the two metal sheets is reduced when a vehicle passes over them. This spacing decrease causes a change in the resonant frequency of the circuit of which the mats are a part. Consequently, by monitoring the resonant frequency of the circuit, the weight of the passing vehicle can be determined [1]. New types of WIM sensors are currently being developed [20][22] and more complex signal processing techniques implemented [23][24][25][26][27] to provide better estimates of vehicle weight and classification capability.

Radar

To detect a vehicle, radars transmit a signal in either the millimeter or microwave regions of the electromagnetic spectrum. When the vehicle passes through the wave, part of the energy is reflected back to a receiver, resulting in detection of the object. The two primary forms of radar used for vehicle detection are Continuous Wave (CW) Doppler and Frequency Modulated Continuous Wave (FMCW). CW Doppler transmits a signal at constant frequency which is then Doppler shifted as it reflects off a moving vehicle. The primary disadvantage to this approach is the inability to detect stationary vehicles [1].

FMCW radar transmits a signal which is constantly changing in time, which allows it to detect stationary vehicles. These radars are used both in forward scanning [28][29][30] and side scanning modes [31][32] depending upon the desired application. Additionally, other forms of radar have also been explored as indicated in [33][34][35].

Ultrasonic Sensors

Ultrasonic sensors emit sound waves which range in frequency between 25 KHz and 50 KHz and measure the reflections from objects within the beam width. The reflected energy allows the sensor to determine the range of the object in the beam. If a range other than the background road is detected, it is counted as a vehicle. An ultrasonic sensor can measure speed by emitting signals at two closely spaced incident angles and recording when the vehicle crosses each beam. Additionally, constant frequency sensors can use the Doppler principle to measure vehicle speed [1]. In addition to downward configurations, ultrasonic sensors can also be used in side looking configurations [36][37][38].

Passive Acoustic Sensors

Passive acoustic sensors detect a vehicle using the acoustic signature produced by the vehicle[1]. Acoustic sensor networks can be implemented both by traditional wired models or by the use of wireless networks [39][40]. The networks can be utilized for detection and speed measurement of individual vehicles [41][42] [43] or be used to infer the general cumulative state of traffic flow [44].

Multimodal Fusion Approaches

Multi-modal type fusion approaches provide the opportunity to fuse multiple forms of information to arrive at an overall better solution. Sensing modalities that are currently being researched include video, audio, seismic, magnetic and infrared[45]. Specifically, visible wavelength cameras are being combined with modalities such as acoustic[46], Doppler radar[47], inductive loop detectors[48] and Infrared cameras[49]. By combining

visible cameras with other modalities, one can mitigate the weaknesses of traditional visible camera, such as the inability to function in the dark and the inability to directly measure range. Additionally, multispectral and hyperspectral information can also be used for classification purposes [50].

Image Based

Image based vehicle detection and classification techniques primarily focus on either background modeling, foreground object detection or a combination of both. Background modeling allows for detection of vehicles by determining a vehicle is not part of the background [51][52]. Foreground object detection utilizes features from the vehicle to separate it from the background and perform additional classification [53]. Typically, the two approaches are coupled for increased performance [54][55]. These techniques can be applied to a single camera or multi-camera system as discussed in [56][57].

Background Modeling

Background modeling can range in complexity depending upon how robust the method is to change in the background. Using background modeling to identify vehicles has the benefit of not being specific to any particular object or object set. The simplest form of background modeling is to use a prior video frame as the background and subtract the earlier frame from the current as shown in [52]. However, this approach is only effective at identifying moving vehicles given that stationary vehicles will be treated as part of the background and subtracted out.

In order to capture the potential variability of a given background pixel, Gaussian Probability Density Functions (PDF) where used to model the mean and variance of background pixels. Using the PDF for both the background and foreground classes, the likelihood of a pixel given each class could be calculated and the image can be segmented appropriately [58][59]. Additionally, if prior probabilities are incorporated, the probability of background given the pixel value and the probability of foreground given the pixel value can each be calculated for image segmentation purposes [51].

Despite the additional descriptive capability provided by Gaussian PDFs for the purpose of discriminating foreground and background pixels, they are still found to be lacking over a wide range of conditions [60]. Consequently, Mixture of Gaussians (MoG) were leveraged to more accurately model the pixel variability as shown in [61] [62]. By using the MoG PDF for both the background and foreground classes, the likelihood of a pixel given each class could be calculated and the image can be segmented appropriately.

In addition to statistical model approaches, Codebook based techniques are also popular for background modelling. Codebook based techniques cluster samples at each pixel and constructs code words to represent the data. In a codebook based approach, each pixel has its own codebook which allows for tremendous flexibility but also increases the complexity of the model [63].

Foreground Object Detection and Classification

Foreground object detection and classification approaches can be broadly grouped in the following two areas: Motion Based Detection and Feature based Detection & Classification. As the name implies, Motion based detection techniques utilize features derived from the motion of the foreground object whereas Feature Detectors extract features from the imagery to detect and classify foreground vehicles.

Motion Based Detection

Motion based detection uses motion based features for detection. Motion based detection only works for vehicles moving above a certain threshold. The simplest form of motion based detection is frame differencing, which subtracts a prior video frame from the current and uses the resulting differences as features for detection [52]. In [54] and [64] the authors utilizes motion vector fields to further discriminate between foreground and background. Another approach shown in [65] demonstrates temporal matching of feature points over multiple frames which is a combination of motion based and feature based detection.

Feature Detection & Recognition

Detecting and classifying vehicles based on features equates to a pattern recognition task and consequently, there are many methods by which it can be achieved. Apart from the actual classification mechanism employed (e.g. SVMs, Neural Networks, Maximum Likelihood) the true distinction between the approaches lies in the choice of features [53]. Feature extraction techniques currently used in vehicle detection and classification include:

Edge & Corner, Histogram Oriented Gradients (HOG), and Transform Domain approaches. Other less common approaches include Deep Learning Convolutional Neural Networks [66] and Scale Invariant Feature Transform (SIFT) [67][68] with various recognition approaches .

Edges and corners are intuitive and powerful features for detection and classification of vehicles. In [65] the author demonstrates the use of the Harris corner detector to detect corners across multiple frames and perform temporal matching between them for the purposes of vehicle detection and tracking. Similarly, in [69] the author uses the Generalized Hough Transform (GHT) to find corners which are used to track a vehicle over time. In [70] the author demonstrates the use of the Sobel transform to find vehicle edges and define bounding boxes for detection and tracking. In [55] background subtraction is combined with edge detection to segment vehicles from the background over varying illumination conditions. Many of the Edge & Corner based techniques would provide viable PoI for the methods presented in this dissertation.

As articulated in [71], HOG features are calculated using the gradient direction histogram of an image. It not only provides both gradient and edge information but also mitigates the effects of varying illumination. The magnitude of the gradient is determined by Equation 1 and the direction by Equation 2.

$$\boldsymbol{m(x,y) = \sqrt{(I(x+1,y) - I(x-1,y))^2 + (I(x,y+1) - I(x,y-1))^2}} \quad (1)$$

$$\boldsymbol{\theta(x,y) = \tan^{-1}\left(\frac{I(x,y+1) - I(x,y-1)}{I(x+1,y) - I(x-1,y)}\right)} \quad (2)$$

Once the gradient is calculated, the image is separated into sub-bins which are then turned into histograms of the gradient values. The histograms are then concatenated into a large feature vectors. The feature vectors are then passed to a boosting [72], SVN [73][74][75][76] or similar type classifiers[77] such that classification can be performed. Though the specific implementation of HOG based approaches varies, many of the approaches found in literature would provide viable PoI for the methods presented in this dissertation.

As is the case with many feature based approaches, implementation in a transform domain is often beneficial and is demonstrated in [78] where Mexican Hat Wavelets are used for adaptive edge detection before being fed to SVM for classification. The most popular transform techniques shown in the literature where the PCA [79][80] and wavelet transforms. The most popular wavelet chosen was the Haar [81][82][83][84]. Regardless of the approach or transform chosen, the basic premise was to use the transform coefficients as feature vectors for classification by an SVM, Ada Boost or similar classifier. Transform domain techniques offer another category of techniques for providing viable PoI for the methods presented in this dissertation.

In general, although there is a great deal of image based tracking work which is applicable to this area, the large majority of it centers on 2D image tracking. The approaches detailed in this work provide a true kinematic representation of the PoI by modelling it in 3D space using inertial tracking principles.

Inertial Tracking

In order to estimate the true position and velocity of the PoI in 3D space such that their respective geometries could be calculated, it is necessary to track the PoI in an Inertial Reference Frame *R* such that the position of the PoI is invariant to the position of the platform sensor. This is accomplished via the implementation of an Extended Kalman Track Filter to optimally estimate the Cartesian Position and Velocity in the Reference Frame *R* using angle measurements measured in the Detector Frame *D*. An Inertial Reference Frame is defined as a frame which is not undergoing any type of acceleration [85].

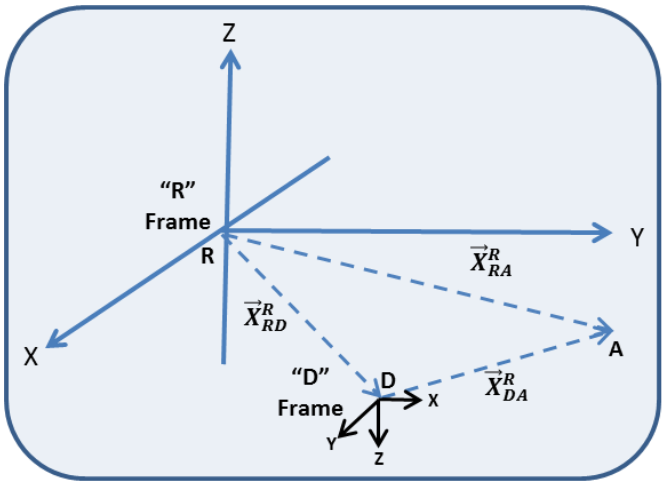

Figure 1 Inertial Track Geometry

The diagram shown in in Figure 1 is a standard Track Geometry from which the Kalman Track Filter Equations can be derived. It represents the positions of the Detector and PoI *A* with respect to the track Reference Frame *R*. Additionally; it represents the orientation of the Detector Frame *D* with respect to the Reference Frame *R*. To transform angles expressed in the Detector Frame *D* to angles expressed in the Reference frame *R*, one must first define the vector in the Detector Frame *D*. This is accomplished by defining a

unit vector aligned with the X-Axis of the Detector Frame *D* and rotating it to point towards PoI *A*. The rotation is accomplished by creating an Euler rotation matrix [86] with the azimuth and elevation angles of PoI *A* relative to the Detector Frame *D* as shown in Equation 3 and Equation 4.

$$\vec{U}_{DA}^{D} = \Phi_D \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{3}$$

$$\Phi_D = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ -\sin\psi & \cos\psi & 0 \\ \sin\theta\cos\psi & \sin\theta\sin\psi & \cos\theta \end{bmatrix} = \Phi[\theta^D]\Phi[\psi^D] \tag{4}$$

After the vector between the Detector and PoI *A* has been defined, it must then be transformed into the Reference Frame *R*. This is accomplished via an Euler Transformation Matrix as shown in Equation 5 and Equation 6.

$$\vec{U}_{DA}^{R} = C_D^R \, \vec{U}_{DA}^{D} \tag{5}$$

$$C_D^R = \begin{bmatrix} \cos\theta\cos\psi & -\sin\psi & \sin\theta\cos\psi \\ \cos\theta\sin\psi & \cos\psi & \sin\theta\sin\psi \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} = C\left[\theta_D^R\right] C\left[\psi_D^R\right] \tag{6}$$

Finally, by using the unit vector between the Detector bore sight and PoI *A* expressed in the *R* frame, the azimuth and elevation angles of the detection can be calculated using Equation 7 and Equation 8.

$$\boldsymbol{\psi_{DA}^{R}} = \mathbf{tan^{-1}}\left(\frac{\vec{U}_{DA_Y}^{R}}{\vec{U}_{DA_X}^{R}}\right) \quad (7)$$

$$\boldsymbol{\theta_{DA}^{R}} = -\mathbf{sin^{-1}}\left(\vec{U}_{DA_Z}^{R}\right) \quad (8)$$

The azimuth and elevation angles between the Detector and PoI *A* expressed in the Reference Frame *R* will be the angular measurement inputs for the Track Filter equations.

Pattern Recognition

In order to achieve the desired vehicle classification, multiple sets of features, represented in three separate domains are evaluated using four separate classification approaches.

Information Representation in Multiple Domains

The efficiency of representing information can vary greatly depending upon the chosen domain and the associated basis vectors employed to represent the information. Representing information more efficiently, i.e. with fewer coefficients, is the foundation of many dimensionality reduction and compression techniques. Transformation from one domain to another is accomplished via a linear transformation wherein all the energy, i.e. the information, of a signal is retained [87].

The Fourier Transform effectively demonstrates the benefits of representing information in alternative domains when the case of a single cosine is considered. A cosine wave in the time domain requires numerous points to capture the structure of the signal.

Alternatively, that same cosine in the Fourier domain is fully characterized by a single coefficient. Given that any time domain signal can be represented as a combination of sines and cosines, the Fourier Domain is often effective at compactly representing very complex time domain signals [87].

Domain transformation for the purpose of dimensionality reduction has multiple benefits when used to preprocess data prior to pattern recognition. By reducing the size of the feature vector presented to a classifier, the processing burden required to do the classification is reduced. There is an upfront processing cost for transforming the data; however the efficiency of many transform techniques mitigates the transformation cost and typically the processing savings on the back end classification far exceeds the transformation costs [88][89]. The fact that each domain represents a given signal differently can be exploited for the purposes of recognition or compression. Fusing of information from multiple domains provides a more robust ensemble classifier than can be obtained by using a single domain representation [90][91]. In this dissertation we consider data in the Native Spatial domain, the PCA domain and the LDA domain.

Domain transformation and by extension dimensionality reduction also addresses what is known as the "Curse of Dimensionality". The "Curse of Dimensionality" refers to the fact the volume of the feature space increases rapidly as a function of the number of dimensions of the feature vector. The feature space increases so rapidly that the available training data becomes sparse as compared to the total volume of the feature space and consequently does not provide an accurate representation of the various classification types [92]. This issue if mitigated by dimensionality reduction because the number of features is

reduced and by extension the volume of the feature space is reduced making the available data less sparse.

Native Spatial Domain

The domain in which the data was collected or initially calculated is referred to as the Native Spatial Domain. Other than the calculation of the features, there are no additional steps required to process the data in the Native Spatial Domain. Consequently, any correlation between features will persist thus allowing for the possibility of redundant information in the feature set. Additionally, classification approaches that assume the features are uncorrelated or independent will also have diminished efficacy.

PCA

PCA provides a mechanism by which the Native Spatial data can be represented in a transform domain where the basis vectors are orthogonal. In other words, PCA attempts to find the most efficient representation of the data, thus removing redundant information in the feature set. PCA assumes the transformation from the Native Spatial domain to the PCA domain is a linear process. Additionally, it is assumed that the distribution of the feature set is Gaussian, thus the mean and variance of the feature set are sufficient statistics for the distribution. Finally, PCA assumes large variances are indicative of the important characteristics of the signal [93]. Given that PCA is a domain transform, it can be used as a pre-processing step before evaluating features in numerous discriminant functions or classifiers. The work in this dissertation applies PCA to the Native Spatial vector angle and magnitude feature set prior to the features being evaluated by the Minimum Euclidean

Distance, Maximum Likelihood, Artificial Neural Network and Bayesian Ensemble classifiers. In general, PCA has numerous applications in a wide array of disciplines including compression [94][95], detection [96][97] and classification [89].

PCA accomplishes feature de-correlation by pre-multiplying a given feature matrix with the transpose of Eigenvectors which were calculated from statistically similar data. The proof below [93] shows that for a given matrix X, feature de-correlation can be achieved by pre-multiplying a given feature matrix with the transpose of Eigenvectors. This is proven by considering the N by M matrix X consisting of N rows of M features. Our goal is to define a matrix A which could transform X into Y, as shown in Equation 9, such that the columns of Y are uncorrelated with one another.

$$\boldsymbol{Y = AX} \tag{9}$$

De-correlation of the columns of Y will be achieved if the non-diagonal elements of the covariance of Y are zero. In order to demonstrate the de-correlation, the covariance of Y can be calculated as shown in Equation 10.

$$\boldsymbol{\Sigma_Y = YY^T} \tag{10}$$

By substituting Equation 9 in Equation 10, Equation 11 expresses the covariance of Y in terms of X and A.

$$\boldsymbol{\Sigma_Y = AXX^TA^T} \tag{11}$$

Substituting the covariance for *X* into Equation 11 leads to the covariance of *Y* in terms of *A* and covariance of *X* as shown in Equation 12.

$$\boldsymbol{\Sigma_Y = A\Sigma_X A^T} \quad (12)$$

Using Eigen decomposition, the covariance of *X* can be factored into Equation 13 were *E* is the orthogonal eigenvector matrix calculated from the covariance of *X* and $\boldsymbol{\Lambda}$ is a diagonal matrix containing the eigenvalues calculated from the covariance of *X*.

$$\boldsymbol{\Sigma = E\Lambda E^{-1}} \quad (13)$$

Given that the inverse of an orthogonal matrix is its transpose and the eigenvector matrix is an orthogonal matrix, equation 14 can be stated.

$$\boldsymbol{E^{-1} = E^T} \quad (14)$$

The substitution of Equations 13 and 14 into Equation 12 will result in equation 15.

$$\boldsymbol{\Sigma_Y = AE_X\Lambda_X E_X^T A^T} \quad (15)$$

Equating the transpose of the eigenvector matrix to *A,* as shown in Equation 16, and substituting into Equation 15 will result in equation 17.

$$\mathbf{A} = \boldsymbol{E_X}^{\boldsymbol{T}} \quad (16)$$

$$\boldsymbol{\Sigma_Y} = \boldsymbol{E_X}^{\boldsymbol{T}} \boldsymbol{E_X} \boldsymbol{\Lambda_X} \boldsymbol{E_X^T} \boldsymbol{E_X} \quad (17)$$

Equation 17 can be simplified by recognizing the multiplication of a matrix by its transpose results in an identity matrix.

$$\boldsymbol{\Sigma_Y} = \boldsymbol{I} \boldsymbol{\Lambda_X} \boldsymbol{I} \quad (18)$$

Furthermore, any matrix multiplied by an identity matrix is unchanged, allowing us to drop the identity matrices from equation 18, thus simplifying to what is shown in equation 19.

$$\boldsymbol{\Sigma_Y} = \boldsymbol{\Lambda_X} \quad (19)$$

Finally, Equation 19 states the covariance of *Y* is equal to the diagonal eigenvalue matrix of *X*. The diagonal covariance matrix proves de-correlation because the off diagonal terms are zero.

LDA

The aim of Linear Discriminant Analysis is to find a domain in which data can be represented such that the likelihood of discrimination is maximized. It achieves this by maximizing the ratio shown in Equation 27, which is the ratio of the "between class scatter"

matrix to the "within class scatter" matrix expressed in the LDA transform domain [98]. This approach is in contrast to PCA which is concerned with the most efficient representation of the data [93]. Though its name implies LDA is a discriminant function, it is in fact a linear transform which can be used as a pre-processing step before evaluating features in numerous discriminant functions or classifiers. The work in this dissertation applies the LDA transform to the Native Spatial vector angle and magnitude feature set prior to the features being evaluated by the Minimum Euclidean Distance, Maximum Likelihood, Artificial Neural Network, Bayes Ensemble classifiers.

The nature of the LDA transform results in a natural dimensionality reduction where the number of resulting dimensions is the number of types being classified minus one [92]. For example, in this dissertation we are evaluating the classification performance of three separate vehicle types. Consequently, applying the LDA transform to the feature set, regardless of whether the feature set contains two, three, seven or sixteen Native Spatial features, will result in an LDA feature set with two LDA features. In general, LDA has been used for dimensionality reduction and classification in numerous applications in a wide array of disciplines [99][100][101][102].

As with any linear transform, the LDA transform is achieved by multiplication of the LDA weight matrix with the Native Spatial features matrix. Equations 20 through 34 demonstrate how to calculate the LDA w matrix. The first step in calculating the LDA w matrix [98] is to calculate the scatter matrix for each type as shown in Equation 20. The scatter matrix is the equivalent of the covariance matrix for that type.

$$\boldsymbol{S_i} = \sum_{\boldsymbol{x}\in\boldsymbol{\omega_i}} (\boldsymbol{x} - \boldsymbol{\mu_i})(\boldsymbol{x} - \boldsymbol{\mu_i})^{\boldsymbol{T}} \quad (20)$$

Step two is to calculate the mean values of each of the features for each type as shown in Equation 21.

$$\boldsymbol{\mu_i} = \frac{\mathbf{1}}{\boldsymbol{N_i}} \sum_{\boldsymbol{x} \in \boldsymbol{\omega_i}} \boldsymbol{x} \qquad (21)$$

Step three is to calculate the "within class scatter" matrix by summing the scatter matrices for each type as shown in Equation 22.

$$\boldsymbol{S_W} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{C}} \boldsymbol{S_i} \qquad (22)$$

Step four is the calculate the overall mean value for all features in all types by taking the mean values for each type, weighting it by the number of samples of that type, summing the results across all types and dividing by the number of types as shown in Equation 23.

$$\boldsymbol{\mu} = \frac{\mathbf{1}}{\boldsymbol{N}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{C}} \boldsymbol{N_i \mu_i} \qquad (23)$$

Step five is to calculate the "between class scatter" matrix as shown in Equation 24.

$$\boldsymbol{S_B} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{C}} \boldsymbol{N_i}(\boldsymbol{\mu_i} - \boldsymbol{\mu})(\boldsymbol{\mu_i} - \boldsymbol{\mu})^{\boldsymbol{T}} \qquad (24)$$

Step six is to represent the "between class scatter" matrix in the LDA transform domain in terms of the Native Spatial domain values and weighting matrix *w* as shown in Equation 25.

$$\tilde{S}_B = w^T S_B w \quad (25)$$

Step seven is to represent the "within class scatter" matrix in the LDA transform domain in terms of the Native Spatial domain values and weighting matrix *w,* as show in in Equation 26.

$$\tilde{S}_W = w^T S_W w \quad (26)$$

Step eight is to define the equation for the ratio of the "between class scatter" matrix to the "within class scatter" matrix expressed in the LDA transform domain but in terms of the Native Spatial domain as shown in Equation 27.

$$J(w) = \frac{\tilde{S}_B}{\tilde{S}_W} = \frac{w^T S_B w}{w^T S_W w} \quad (27)$$

Step ten is to maximize the ratio of the "between class scatter" matrix to the "within class scatter". To do so, we must take the derivative of Equation 27 with respect to the weighting matrix *w* and set it equal to zero as shown in Equation 28 through Equation 31.

$$\frac{d}{dw}[J(w)] = \frac{d}{dw}\left[\frac{w^T S_B w}{w^T S_W w}\right] = 0 \quad (28)$$

$$[w^T S_W w]\frac{d[w^T S_B w]}{dw} - [w^T S_B w]\frac{d[w^T S_W w]}{dw} = 0 \quad (29)$$

$$[\boldsymbol{w^T S_W w}]\boldsymbol{2S_B w} - [\boldsymbol{w^T S_B w}]\boldsymbol{2S_W w} = \boldsymbol{0} \quad (30)$$

$$\left[\frac{\boldsymbol{w^T S_W w}}{\boldsymbol{w^T S_W w}}\right]\boldsymbol{S_B w} - \left[\frac{\boldsymbol{w^T S_B w}}{\boldsymbol{w^T S_W w}}\right]\boldsymbol{S_W w} = \boldsymbol{0} \quad (31)$$

Step eleven is to Simplify Equation 31 by cancelation and substitution resulting in Equation 32.

$$\boldsymbol{S_B w - J S_W w = 0} \quad (32)$$

Step twelve is to re-organize Equation 32 such that it can be expressed as the generalized Eigenvalue equation shown in Equation 33.

$$\boldsymbol{{S_W}^{-1} S_B w = J w} \quad (33)$$

Step thirteen is to solve the generalized Eigenvalue equation as show in Equation 34 such that the Eigenvector matrix can be calculated. The *w* matrix is equivalent to the matrix of the non-trivial Eigenvectors.

$$\boldsymbol{\left({S_W}^{-1} S_B - JI\right) w = 0} \quad (34)$$

Step fourteen, the final step, is to multiply the Native Spatial feature matrix by the transpose of the *w* matrix which will transform the Native Spatial domain features into the LDA domain.

Classifiers

Four classification approaches where chosen to demonstrate the classification & identification capabilities of the approach: Minimum Euclidean Distance, Maximum Likelihood, Artificial Neural Networks (ANN) and Bayesian Ensemble. Each classifier can be evaluated in multiple transform domains.

Minimum Euclidean Distance

The first classifier, Minimum Euclidean Distance, also referred to as a minimum distance classifier, utilizes the first order statistics, or mean value, of the feature vector to infer a classification or identification. It is assumed that the mean value of a feature vector for a given vehicle type is the "Truth" data for that given vehicle type.

$$\boldsymbol{D(X_i, C_n) = \sqrt{\sum_{i=1}^{K}(X_i - \mu_{c_n})^2}} \quad (35)$$

Classification and or identification is achieved by comparing the unknown feature vector X_i to the truth value μ_{c_i} of each class. The class which results in the smallest Euclidean distance between the unknown feature vector X_i and the truth value μ_{c_i} is selected as the vehicle type of the unknown feature vector [103].

Maximum Likelihood

The second classifier, Maximum Likelihood, utilizes the first and second order statistics, mean and variance, of the feature vector to infer a classification or identification. It

is assumed that the mean value of a feature vector for a given vehicle type is the "Truth" data for that given vehicle type. Furthermore, the variance for a given feature vector for a given vehicle type is calculated by calculating the variance among the training samples generated by the simulation. Maximum likelihood classification operates under the assumption that a feature from a given class will achieve its highest likelihood when evaluated in the PDF which is representative of its true class [92].

$$\boldsymbol{\mathcal{L}(X_i) = \frac{1}{\sqrt{(2\pi)^2|\Sigma_{C_n}|}} exp\left(-\frac{1}{2}\left(X_i - \mu_{C_n}\right)^T \Sigma_{C_n}^{-1}\left(X_i - \mu_{C_n}\right)\right)} \quad (36)$$

In this disseration, it is assumed that each feature will follow a Gaussian distribution with the calculated variance centered around the the true mean value. Classification and or identificaiton is achieved by calculating a Gaussian distribution for each vehicle type and then calculating the Likelihood the unknown feature vector X_i is part of the the vehicle type distribution. The vehicle type distribution which results in the highest likelihood is selected as the vehicle type of the unknown feature vector X_i. An alternative formulation for this classifier could have calculated the log likelihood of a distribution.

Feed Forward Artificial Neural Networks

Unlike Euclidean Distance classifiers, feed forward Artificial Neural Networks have the ability to utilize non-linear decision boundaries. This is accomplished by a multi-layer neural network which first projects the non-linear features into a domain where they are linearly separable and then finds a linear boundary to perform classification [92]. In this

Dissertation, a feed forward neural network classifier was trained and implemented to assess each set of features in each domain with the goal of vehicle classification. Neural Networks are used for various applications over multiple disciplines and problem domains. Applications include license plate recognition [104], traffic incident detection [105], pedestrian detection [106], vehicle classification [107][108] and facial recognition [109].

Bayes Ensemble Classifier

Bayesian Networks allow Subject Matter Experts (SMEs) to embed domain specific knowledge into the information fusion process. They provide a scalable mechanism by which complex relationships between random variables can be modeled and exploited for the purposes of information fusion. One of the primary tasks for Bayesian Networks when applied to the problem of information fusion is to infer the value of an unknown or hidden state by using the values of observable states, i.e. states that can be measured. This is done by leveraging a posteriori knowledge, or knowledge gained through experience [110], of the relationships between the hidden state and the observable state. In other words, by using the knowledge of the relationship between the two states, knowing information about one state allows you to infer information about the other state [92]. Bayesian Networks are used for various applications over multiple disciplines and problem domains. Applications include sensor fusion [111], wireless networking [112], resource allocation [113], speech processing [114] and classification [115][116][117][118][119].

There are many methods by which Bayesian Networks can achieve information inference. The two primary approaches are exact and inexact methods. With exact methods, such as variable elimination or junction trees, one computes the conditional probability

distributions analytically over the various states. However, when the computational complexity of exact inference becomes too great, we turn to inexact or approximation methods which trade computation time for accuracy [120].

To provide an intuitive explanation of the process of information inference using Bayesian Networks, the variable elimination approach [121] is demonstrated in Figure 2 below. The example demonstrates the steps of combining and marginalizing distributions to arrive at the desired conditional probability. Sub-Diagram A of Figure 2 demonstrates a simple Bayesian Network with four nodes or variables. For the purpose of this explanation, node *X* will be the true type of an object of interest which cannot be directly measured. The fact it cannot be directly measured makes it a hidden node. Nodes *A* and *B* both represent the output of two separate classifiers attempting to determine *X*. Both *A* and *B* can be directly measured. Node *R*, which can be measured, represents the range of the object represented by *X*. The probability of nodes *A* and *B* correctly identifying the object represented by *X* is dependent on both the value of *X* and the value of *R*.

To capture the a posteriori relationship between *A*, *B*, *X* and *R*, conditional distributions are calculated using historical data. If the conditional distributions are discrete, they are referred to as Conditional Probability Tables (CPTs) .The conditional distribution for node *A* is represented by the expression $P(A|XR)$ and the conditional distribution for node *B* by the expression $P(B|XR)$. Given that neither *X* nor *R* have parent nodes, their values are not dependent on any other variable allowing their prior probabilities to be represented by the expressions $P(X)$ and the $P(R)$ respectively.

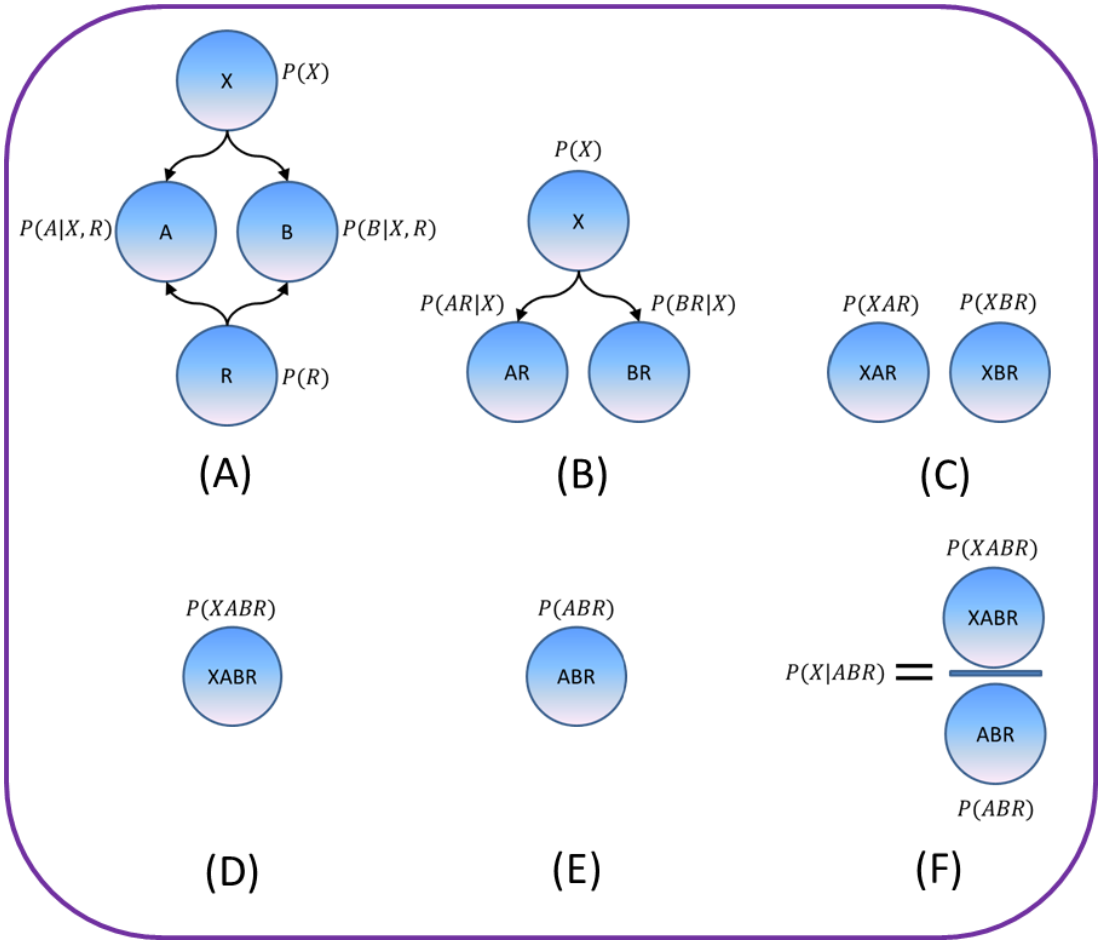

Figure 2 Bayesian Inference

The values of either *A*, *B* or *R* can be used to infer information about the value of *X*. For the purposes of this explanation, it is assumed the values of *A*, *B* and *R* are known and the goal is to calculate the conditional distribution which represents the probability of *X* given *A*, *B* and *R* as represented by the expression $P(X|ABR)$. Step one of the information inference is to join node *A* with node *R*. This is accomplished by multiplying the conditional distribution $P(A|XR)$ with the prior probability $P(R)$ resulting in $P(AR|X)$. It should be noted that there are many valid orders in which the nodes could be combined. Step two is to

multiply the conditional distribution $P(B|XR)$ with the prior probability $P(R)$ resulting in $P(BR|X)$. Performing those two joins will result in the new network shown in sub-diagram *B*. Step three is then to join $P(AR|X)$ with $P(X)$ resulting in in joint probability $P(XAR)$. Similarly, we must join $P(BR|X)$ with $P(X)$ resulting in $P(XBR)$ which creates the new network shown in Figure 2 sub-diagram *C*. Step four is to multiply together $P(XAR)$ with $P(XBR)$ resulting in $P(XABR)$ as shown in Figure 2 sub-diagram *D*. Step five is to sum out, or marginalize, $P(XABR)$ with respect to *X* resulting in the joint distribution $P(ABR)$ as shown in Figure 2 sub-diagram *E*. Finally, step six is to divide $P(XABR)$ by $P(ABR)$ resulting in the conditional distribution $P(X|ABR)$ as shown in Figure 2 sub-diagram *F*. The expression $P(X|ABR)$ represents the probability for a given value of *X* conditional upon the observable states *A*, *B* and *R*.

Figure 3 depicts the structure of the Bayes Ensemble Classifier used in this dissertation. The structure implies the output of the LDA classifier and PCA classifier are both dependent upon the true vehicle type and the range of the vehicle from the airborne platform.

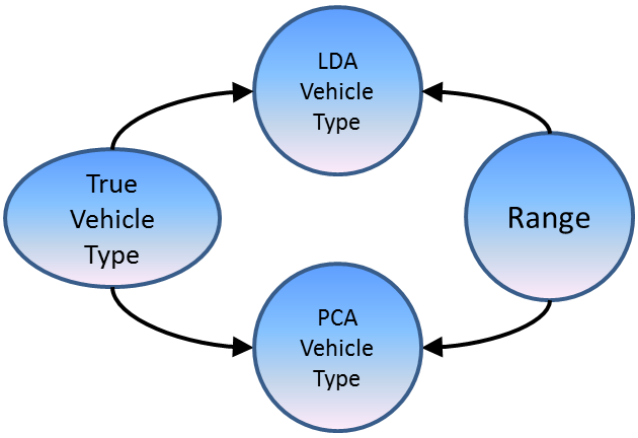

Figure 3 Bayes Ensemble Classifier

The a posteriori relationship between the true vehicle type, Range and a classifier is expressed through the use of a Conditional Probability Table (CPT). The CPT can be calculated by assembling a confusion matrix from a classifiers prior performance as a function of range and true type and then normalizing it. Using the CPT and the prior probabilities, the output from both the LDA and PCA classifiers can be fused into a single ensemble classifier. This process is repeated for each classification approach, each filter type and each level of noise.

CHAPTER THREE:
IMPLEMENTATION

To demonstrate the efficacy of the proposed approaches for vehicle classification, the following system was implemented in Matlab using high fidelity physics based simulation. An aerial platform (such as a small UAV, plane or helicopter) with a stabilized passive imaging system having an Instantaneous Field Of View (IFOV) of 50 micro-radians was simulated at varying ranges, altitudes and angular geometries with respect to three types of vehicles. The vehicles were traveling at a constant velocity towards the aircraft. Figure 4 depicts a placement grid for each of the vehicles for a single aircraft altitude. The placement grid varied from ± 2° in azimuth with increments of 1° and 200 meters (m) to 600 meters in range with increments of 100 meters. There were a total of 25 possible positions. The altitude of the craft is determined by the desired platform to vehicle elevation angle. The elevation angle varied from 1° to 4° with increments of 1°. During the simulation, five of each vehicle type are placed at each of the grid positions and moved towards the aerial platform at a constant velocity. This is repeated for each of the desired elevation angles.

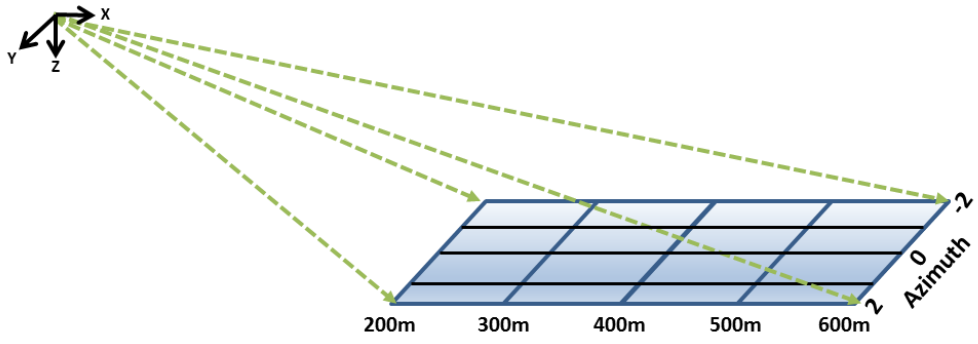

Figure 4 Vehicle Placement

For each of the variants of vehicle position, type and motion the aerial platform maintains a constant altitude as determined by the desired initial engagement elevation angle and moves towards the vehicles. The aircraft performs a maneuver similar to a sine wave in which it perturbs its elevation and azimuth position relative to the vehicle of interest. This maneuver is required in order to give the necessary observability via inference into the unobservable states of the track filter. Each engagement in the simulation ran for six seconds with a frame update rate of 30Hz.

In a non-simulated implementation, each frame of imagery would be fed to an image processing algorithm, which would then extract the PoI from the imagery. The angular measurement of the PoI would then be sent to the inertial tracker such that there 3D positions could be inferred. Given that our approach is independent of any specific front end image processing algorithms, the aerial platform and the PoI are modelled as centroids in 3D Earth Centered Earth Fixed (ECEF) inertial space. To achieve the correct PoI placement, an ECEF position and orientation is chosen for the aerial platform and the PoI are placed relative to the aerial platform sensor, such that the desired engagement geometry is achieved as shown Figure 5A. The absolute ECEF position of the PoI can then be calculated by adding the absolute ECEF position of aerial platform to the relative vectors between the aerial platform and the PoI as shown in Figure 5B.

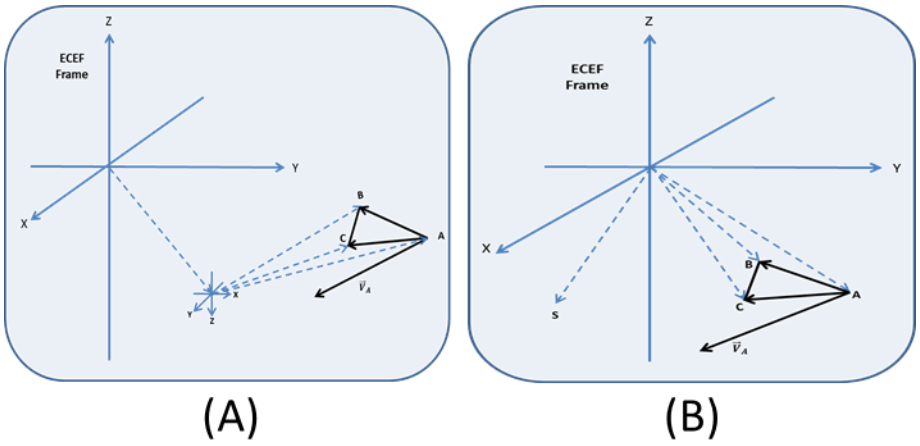

Figure 5 Aerial Platform & Vehicle ECEF Placement

The physical characteristics chosen for the vehicle PoI were the top center of the windshield, passenger side headlight and drivers side mirror. The 3D geometry of the PoI where modeled as having a rigid geometry to one another to match the rigid geometry of the PoI on the vehicle. Once the 3D PoI for each vehicle type where placed at the correct location and orientation for each of the various runs, they were transformed from a 3D position to the 2D angular measurement (azimuth and elevation) as would be seen by the stabilized passive imaging system.

In order to increase the fidelity of the simulation and assess the robustness of the approach to noise, four separate levels of noise where added to each of the azimuth and elevation measurements before they were passed to the tracker. The noise was modelled as zero mean Gaussian noise with a standard deviation equivalent to one of the following: 50 micro-radians, 100 micro-radians, 150 micro-radians and 200 micro-radians. The noise levels corresponded to 1, 2, 3 or 4 IFOVs respectively. It was assumed all angular noise was uncorrelated with one another.

The simulation software architecture used to implement the approach detailed above is shown in Figure 6 and can be split into three main components: Platform to Vehicle Geometry Generation, Tracking & Feature Calculation and Classification. In order to ensure the generality of the proposed approaches are adequately tested, separate sets of training and test data were generated for the purposes of classifier training and testing. The large majority of the software in the simulation was written in the course of this research; however the evaluation of the Bayesian networks was performed using the Bayes Net Toolbox For Matlab [122] and the training and evaluation of the feed forward Neural Networks was done using the Matlab Neural Network Toolbox [123].

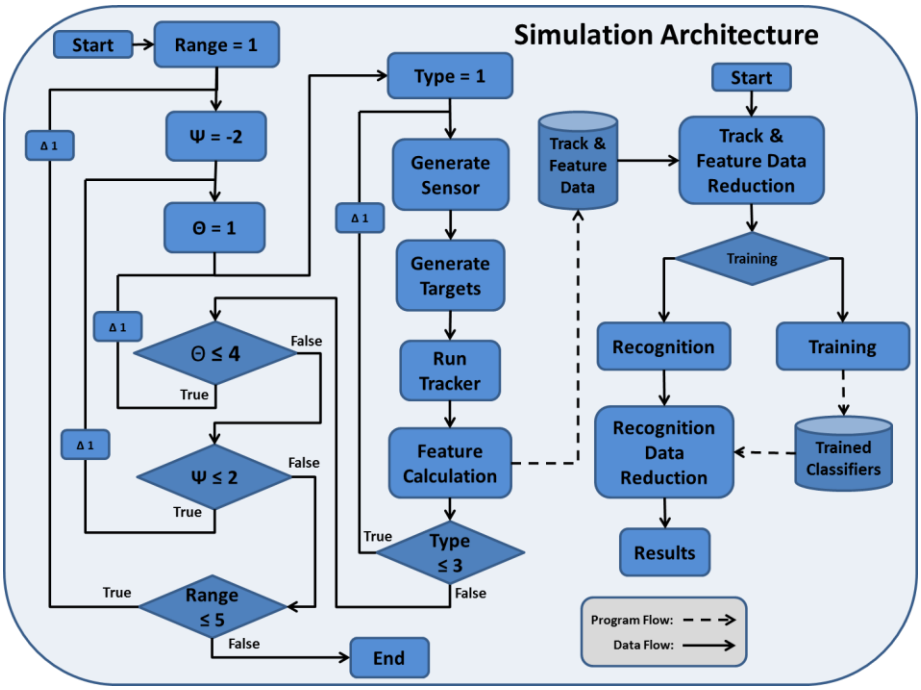

Figure 6 Simulation Software Architecture

Vehicle Types

The three vehicle types chosen for the simulation were a Toyota Sienna Minivan, a Toyota Camry Sedan, and a Toyota Tundra pickup truck. The unique physical dimensions and geometries of each of the vehicle types was measured and modeled in the simulation. Table 2 lists the magnitudes and Euler angular relationships of the vectors which connect the three PoI as represented in the Velocity Frame V. Velocity Frame V is defined as follows. The X-Axis of the reference frame V will be aligned with the velocity vector of PoI A on the vehicle. The Y-Axis and Z-Axis will be calculated using cross product operations and when combined with the X-Axis will result in a right handed coordinate system to serve as reference frame V. The vector lengths and angles listed in Table 2 are considered to be the "Truth" for the purposes of vehicle classification.

Table 2
Vehicle Type Features in "V" Frame

Type	$\boldsymbol{\theta_{AB}}$	$\boldsymbol{\psi_{AB}}$	$\boldsymbol{r_{AB}}$	$\boldsymbol{\theta_{AC}}$	$\boldsymbol{\psi_{AC}}$	$\boldsymbol{r_{AC}}$	$\boldsymbol{\theta_{BC}}$	$\boldsymbol{\psi_{BC}}$	$\boldsymbol{r_{BC}}$
1 Sienna	-21.3°	25.8°	1.82(m)	-21.3°	-76.6°	1.12(m)	6.69°	-126.2°	2.19(m)
2 Camry	-16.5°	21.4°	1.96(m)	-19.0°	-68.2°	1.01(m)	6.21°	-131.6°	2.11(m)
3 Tundra	-17.9°	21.1°	2.49(m)	-14.0°	-84.1°	1.26(m)	8.88°	-135.24°	2.97(m)

Figure 7 Vehicle Type 1- Sienna - Forward

Figure 8 Vehicle Type 1 – Sienna - Side

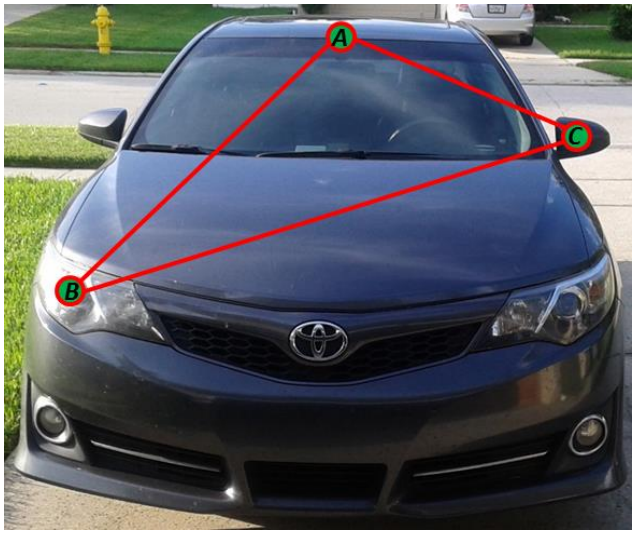

Figure 9 Vehicle Type 2 – Camry - Forward

Figure 10 Vehicle Type 2 – Camry - Side

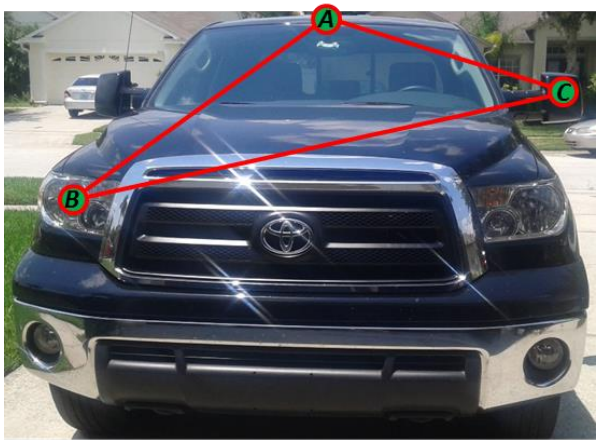

Figure 11 Vehicle Type 3 – Tundra - Forward

Figure 12 Vehicle Type 3 - Tundra - Side

Kalman Filter

Kalman Filters are Bayesian Filters (aka Dynamic Bayesian Networks) that integrate measurable observations (aka measurements) over time to infer the values of latent or hidden states that can't be directly measured. The distribution of the measurable state is dependent upon the value of the hidden state. Consequently, a priori knowledge of the relationship between the hidden and measurable states along with knowledge of how the hidden state distribution evolves over time allows for the inference of the hidden state distribution using measurements from the measurable states. By iterating between prediction and measurement, the estimate of the posterior distribution of the hidden state is improved, allowing the filter to converge upon the true distribution for the hidden state. Although a common use for Kalman Filters are tracking applications, the ability to integrate multiple measurements over time to infer information provides a powerful tool for the purposes of classification, identification and information fusion. For the work in this dissertation, the Kalman filter will be used to estimate the position and velocity information for the PoI, such that the required angle and magnitude features can be calculated and used for classification purposes.

Kalman Filters make the following simplifying assumptions to improve computational efficiency. The Kalman filter assumes the state and measurement noise distributions are normal (Gaussian) distributions. Additionally, it assumes the a priori model which relates the measurable state to the hidden state is a linear relationship, meaning that it can be accomplished via matrix multiplication. Additionally, it assumes the model which predicts the evolution of the hidden state over time is also a linear relationship. If these assumptions are correct, the Kalman filter provides the optimal solution for estimation of

the states. By making these assumptions, calculating the posterior distribution for the unknown state is a fairly straight forward algebraic operation which can be efficiently implemented using matrix operations in modern processors [124][125]. It's for this reason, Kalman filters have found such ubiquitous application for multiple engineering applications. Unfortunately, the Gaussian and linear assumption are not always correct which lead to sub-optimal estimation performance. In the context of a standard track filter, the assumption is the kinematic laws that govern the motion of a target are a set of linear equations. That assumption is correct if a Cartesian based tracker is being used. However, the next assumption is the relationship between the angular measurements and the Cartesian track states is linear. That assumption is incorrect. In order to mitigate the linearity issue with Kalman filters, Extended Kalman filters are typically implemented. Essentially, the non-linear relationships are linearized by approximating the non-linear equations with first order Taylor series approximations. These approximations are valid as long as the derivative of the non-linear equation is linear for small changes [124][125]. An additional issue with Extended Kalman filters is the level of complexity required to linearize the non-linear equations can become prohibitive to implementation. Regardless, Kalman filters and their extended brethren provide a powerful tool for estimation of information that is not directly measurable thus providing a value state estimation tool.

As indicated, the Kalman filter achieves estimation using iterative predication, measurement and update steps. Consequently, a predicted state vector can be calculated by propagating the current state estimate forward in time using the state transition matrix, as shown in Equation 38.

$$\widehat{\boldsymbol{X}} = \begin{bmatrix} \widehat{\boldsymbol{X}}_{\boldsymbol{DT}}^{\boldsymbol{R}} \\ \widehat{\boldsymbol{V}}_{\boldsymbol{DT}}^{\boldsymbol{R}} \end{bmatrix} \quad (37)$$

$$\widehat{\boldsymbol{X}}_{\boldsymbol{k|k-1}} = \mathbf{F}_{\boldsymbol{k}} \widehat{\boldsymbol{X}}_{\boldsymbol{k-1}} \quad (38)$$

Additionally, the State Covariance must also be propagated forward in time using the state transition matrix as shown in Equation 39.

$$\boldsymbol{P}_{\boldsymbol{k|k-1}} = \mathbf{F}_{\boldsymbol{k}} \boldsymbol{P}_{\boldsymbol{k-1}} \mathbf{F}_{\boldsymbol{k}}^{\boldsymbol{T}} \quad (39)$$

After the prediction step, the update step is performed by taking an angular measurement of the PoI and calculating a residual. The residual is calculated as shown in Equation 40 by mapping the propagated state estimate into measurement space using the measurement mapping matrix and then subtracting it from the measurement vector.

$$\boldsymbol{y}_{\boldsymbol{k}} = \boldsymbol{z}_{\boldsymbol{k}} - \boldsymbol{H}_{\boldsymbol{k}} \widehat{\boldsymbol{X}}_{\boldsymbol{k|k-1}} \quad (40)$$

Once the residual is obtained, the Kalman Gain matrix can be calculated as shown in Equation 41 and Equation 42 in order to perform the Update step.

$$\boldsymbol{S}_{\boldsymbol{k}} = \mathbf{H}_{\boldsymbol{k}} \boldsymbol{P}_{\boldsymbol{k|k-1}} \mathbf{H}_{\boldsymbol{k}}^{\boldsymbol{T}} \quad (41)$$

$$\boldsymbol{K}_{\boldsymbol{k}} = \boldsymbol{P}_{\boldsymbol{k|k-1}} \mathbf{H}_{\boldsymbol{k}}^{\boldsymbol{T}} \boldsymbol{S}_{\boldsymbol{k}}^{\boldsymbol{-1}} \quad (42)$$

Once the Kalman gain is calculated as shown in Equation 42, the Update step can be performed as shown in Equation 43.

$$\widehat{\boldsymbol{X}}_{\boldsymbol{k|k}} = \widehat{\boldsymbol{X}}_{\boldsymbol{k|k-1}} + \boldsymbol{K_k y_k} \quad (43)$$

Finally, the Covariance must be updated with the latest information from the measurement as shown in Equation 44.

$$\boldsymbol{P_{k|k}} = (\boldsymbol{I} - \boldsymbol{K_k H_k})\boldsymbol{P_{k|k-1}} \quad (44)$$

Six State Track Filter Derivation

The first group of proposed approaches utilizes separate six state track filters to independently track two or three PoI. To do so, an Extended Kalman Filter was implemented to infer Cartesian position and velocity from angle measurement. The derivation of the filter shown in Equations 45 through Equation 74 follows the general derivation outlined in [124][126][127] with modifications made to fit the problem at hand. The first step of the derivation of the Six State filter is to define the filter state as shown in Equation 45.

$$\widehat{\boldsymbol{X}} = \begin{bmatrix} \widehat{\boldsymbol{X}}_{\boldsymbol{RA}}^{\boldsymbol{R}} \\ \widehat{\boldsymbol{V}}_{\boldsymbol{RA}}^{\boldsymbol{R}} \end{bmatrix} \quad (45)$$

The predicted state estimate for the next time step can be computed by applying the state transition function to the current state estimate $\hat{X}$. Alternatively, the predicted state is equivalent to the state transition function applied to the true state $\bar{X}$ plus the Process Noise vector *w*, as shown in Equation 46.

$$\hat{\boldsymbol{X}}_{\boldsymbol{k|k-1}} = \boldsymbol{f}\left(\hat{\boldsymbol{X}}_{\boldsymbol{k-1}}\right) = \boldsymbol{f}(\bar{\boldsymbol{X}}_{\boldsymbol{k-1}}) + \boldsymbol{w} \tag{46}$$

The Process Noise vector *w* is defined as shown in Equation 47.

$$\boldsymbol{w} = \begin{bmatrix} \boldsymbol{\sigma}_{X_x} \\ \boldsymbol{\sigma}_{X_y} \\ \boldsymbol{\sigma}_{X_z} \\ \boldsymbol{\sigma}_{V_x} \\ \boldsymbol{\sigma}_{V_y} \\ \boldsymbol{\sigma}_{V_z} \end{bmatrix} \tag{47}$$

The Covariance, Q, is defined as the Expected Value of the Process Noise vector *w* multiplied by the transpose of the Process Noise vector *w,* as shown in Equation 48.

$$\boldsymbol{Q} = \mathbf{E}\{\boldsymbol{w}\boldsymbol{w}^{\boldsymbol{T}}\} \tag{48}$$

In order to calculate the residual required for the update step of the filter, the state estimate $\hat{X}$ must be mapped to measurement space. The mapping from state space to measurement space is achieved by using the measurement mapping equation as shown in Equation 49.

Alternatively, the predicted measurement z is equivalent to the mapping of the true state vector $\bar{X}$ to measurement space plus the measurement noise vector v.

$$\boldsymbol{z_{k|k-1} = h\left(\hat{X}_{k|k-1}\right) = h\left(\bar{X}_{k|k-1}\right) + v} \quad (49)$$

The measurement noise vector v is defined as shown in Equation 50

$$\boldsymbol{v = \begin{bmatrix} \sigma_{\psi} \\ \sigma_{\theta} \end{bmatrix}} \quad (50)$$

The Measurement Covariance, R, is defined as the Expected Value of the measurement noise vector v multiplied by the transpose of the measurement noise vector v as shown in Equation 51.

$$\boldsymbol{R = \mathrm{E}\{vv^T\}} \quad (51)$$

As indicated earlier, the predicted state estimate for the next time step can be computed by applying the state transition function to the current state estimate $\hat{X}$. If the state transition function is linear, it can be replaced with a matrix multiplication as shown in Equation 52.

$$\boldsymbol{\begin{bmatrix} \hat{X}^R_{RA_k} \\ \hat{V}^R_{RA_k} \end{bmatrix} = F \begin{bmatrix} \hat{X}^R_{RA_{k-1}} \\ \hat{V}^R_{RA_{k-1}} \end{bmatrix}} \quad (52)$$

The matrix F is computed by deriving the Taylor Series approximation of the state transition functions as show in Equation 53 through Equation 55.

$$\boldsymbol{\Phi} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{53}$$

$$\mathbf{F} = \mathbf{I} + \boldsymbol{\Phi} * \mathbf{\Delta t} \tag{54}$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{\Delta t} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{\Delta t} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{\Delta t} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{55}$$

The measurement vector is shown in Equation 56.

$$\bar{\boldsymbol{z}} = \begin{bmatrix} \boldsymbol{\psi}_{\boldsymbol{DA}}^{\boldsymbol{R}} \\ \boldsymbol{\theta}_{\boldsymbol{DA}}^{\boldsymbol{R}} \end{bmatrix} \tag{56}$$

In order to correctly apply the residual differences to the state estimate and covariance, the measurement mapping matrix must be computed. The measurement mapping matrix quantifies how errors in the measurement vector propagate to errors in the state estimate and state covariance. The measurement mapping matrix is computed by perturbing the measurement vector with respect to the state vector. Consequently, the relationship between

changes in the state vector and changes in the measurement vector is quantified in Equation 57.

$$\boldsymbol{\delta \bar{z}} = \frac{\boldsymbol{\delta z}}{\boldsymbol{\delta \bar{X}}} * \boldsymbol{\delta \bar{X}} + \boldsymbol{v} \quad (57)$$

To linearize the relationship, the Jacobian of the measurement vector is calculated with respect to the state vector, which results in the measurement mapping matrix shown in Equation 59.

$$\boldsymbol{\delta \bar{z}} = \boldsymbol{H} * \boldsymbol{\delta \bar{X}} + \boldsymbol{v} \quad (58)$$

$$\boldsymbol{H} = \begin{bmatrix} \frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{RA_X}^{R}} & \frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{RA_Y}^{R}} & \frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{RT_z}^{R}} & \frac{\delta \psi_{DA}^{R}}{\delta \vec{V}_{RA_X}^{R}} & \frac{\delta \psi_{DA}^{R}}{\delta \vec{V}_{RA_Y}^{R}} & \frac{\delta \psi_{DA}^{R}}{\delta \vec{V}_{RA_Z}^{R}} \\ \frac{\delta \theta_{DA}^{R}}{\delta \vec{X}_{RA_X}^{R}} & \frac{\delta \theta_{DA}^{R}}{\delta \vec{X}_{RA_Y}^{R}} & \frac{\delta \theta_{DA}^{R}}{\delta \vec{X}_{RA_z}^{R}} & \frac{\delta \theta_{DA}^{R}}{\delta \vec{V}_{RA_X}^{R}} & \frac{\delta \theta_{DA}^{R}}{\delta \vec{V}_{RA_Y}^{R}} & \frac{\delta \theta_{DA}^{R}}{\delta \vec{V}_{RA_Z}^{R}} \end{bmatrix} \quad (59)$$

Equations 60 through 74 articulate the calculations required to fully specify the *H* matrix.

$$\boldsymbol{\psi_{DA}^{R}} = \mathbf{tan^{-1}}\left(\frac{\vec{X}_{DA_Y}^{R}}{\vec{X}_{DA_X}^{R}}\right) \quad (60)$$

Partial derivative of ψ_{DA}^{R} with respect to $\vec{X}_{RA}^{R}$:

$$\frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{RA_X}^{R}} = \frac{-\vec{X}_{DA_Y}^{R}}{\left(\vec{X}_{DA_X}^{R}\right)^2 + \left(\vec{X}_{DA_Y}^{R}\right)^2} \quad (61)$$

$$\frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{RA_Y}^{R}} = \frac{\vec{X}_{DA_X}^{R}}{\left(\vec{X}_{DA_X}^{R}\right)^2 + \left(\vec{X}_{DA_Y}^{R}\right)^2} \quad (62)$$

$$\frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{RA_Z}^{R}} = \mathbf{0} \quad (63)$$

Partial derivative of ψ_{DA}^{R} with respect to $\vec{V}_{RA}^{R}$:

$$\frac{\delta \psi_{DA}^{R}}{\delta \vec{V}_{RA_X}^{R}} = \frac{\delta \psi_{DA}^{R}}{\delta \vec{V}_{RA_Y}^{R}} = \frac{\delta \psi_{DA}^{R}}{\delta \vec{V}_{RA_Z}^{R}} = \mathbf{0} \quad (64)$$

Calculate the magnitude r_{DA}:

$$\boldsymbol{r_{DA}} = \sqrt{\left(\vec{\boldsymbol{X}}_{\boldsymbol{DA_X}}^{\boldsymbol{R}}\right)^2 + \left(\vec{\boldsymbol{X}}_{\boldsymbol{DA_Y}}^{\boldsymbol{R}}\right)^2 + \left(\vec{\boldsymbol{X}}_{\boldsymbol{DA_Z}}^{\boldsymbol{R}}\right)^2} \quad (65)$$

Partial derivative of r_{DA} with respect to $\vec{X}_{RA}^{R}$:

$$\frac{\delta r_{DA}}{\delta \vec{X}_{RA_X}^{R}} = \frac{\vec{X}_{DA_X}^{R}}{r_{DA}} \quad (66)$$

$$\frac{\delta r_{DA}}{\delta \vec{X}_{RA_Y}^{R}} = \frac{\vec{X}_{DA_Y}^{R}}{r_{DA}} \quad (67)$$

$$\frac{\delta r_{DA}}{\delta \vec{X}^R_{RA_Z}} = \frac{\vec{X}^R_{DA_Z}}{r_{DA}} \quad (68)$$

Partial derivative of r_{DA} with respect to $\vec{V}^R_{RA}$:

$$\frac{\delta r_{DA}}{\delta \vec{V}^R_{RA_X}} = \frac{\delta r_{DA}}{\delta \vec{V}^R_{RA_Y}} = \frac{\delta r_{DA}}{\delta \vec{V}^R_{RA_Z}} = \mathbf{0} \quad (69)$$

To calculate the value of θ^R_{DA}:

$$\boldsymbol{\theta^R_{DA}} = -\mathbf{sin^{-1}}\left(\frac{\vec{X}^R_{DA_Z}}{r_{DA}}\right) \quad (70)$$

Partial derivative of θ^R_{DA} with respect to $\vec{X}^R_{RA}$:

$$\frac{\delta \theta^R_{DA}}{\delta \vec{X}^R_{RA_X}} = \frac{\vec{X}^R_{DA_Z} * \vec{X}^R_{DA_X}}{r^3_{DA}\sqrt{1-\left(\frac{\vec{X}^R_{DA_Z}}{r_{DA}}\right)^2}} \quad (71)$$

$$\frac{\delta \theta^R_{DA}}{\delta \vec{X}^R_{RA_Y}} = \frac{\vec{X}^R_{DA_Z} * \vec{X}^R_{DA_Y}}{r^3_{DA}\sqrt{1-\left(\frac{\vec{X}^R_{DA_Z}}{r_{DA}}\right)^2}} \quad (72)$$

$$\frac{\delta \theta^R_{DA}}{\delta \vec{X}^R_{RA_Z}} = \frac{-r^2_{DA}+\left(\vec{X}^R_{DA_Z}\right)^2}{r^3_{DA}\sqrt{1-\left(\frac{\vec{X}^R_{DA_Z}}{r_{DA}}\right)^2}} \quad (73)$$

Partial derivative of θ_{DA}^{R} with respect to $\vec{V}_{RA}^{R}$:

$$\frac{\delta\theta_{DA}^{R}}{\delta\vec{V}_{RA_X}^{R}} = \frac{\delta\theta_{DA}^{R}}{\delta\vec{V}_{RA_Y}^{R}} = \frac{\delta\theta_{DA}^{R}}{\delta\vec{V}_{RA_Z}^{R}} = \mathbf{0} \tag{74}$$

Nine State Filter Derivation

Use of a six state track filter requires that each image point of interest be tracked independently of one another. Such an approach does not utilize the known geometric relationship between the points. By tracking the two point geometry in a single filter, the a priori knowledge of the rigid geometry can be used to improve the state estimate and the classification results. The first step of the derivation of the Nine State filter is to define the filter state as shown in Equation 75.

$$\hat{\boldsymbol{X}} = \begin{bmatrix} \hat{\boldsymbol{X}}_{\boldsymbol{RA}}^{\boldsymbol{R}} \\ \hat{\boldsymbol{V}}_{\boldsymbol{RA}}^{\boldsymbol{R}} \\ \hat{\boldsymbol{X}}_{\boldsymbol{AB}}^{\boldsymbol{R}} \end{bmatrix} \tag{75}$$

The predicted state estimate for the next time step can be computed by applying the state transition function to the current state estimate $\hat{X}$. Alternatively, the predicted state is equivalent to the state transition function applied to the true state $\bar{X}$ plus the Process Noise vector *w*, as shown in Equation 76.

$$\hat{\boldsymbol{X}}_{\boldsymbol{k|k-1}} = \boldsymbol{f}\left(\hat{\boldsymbol{X}}_{\boldsymbol{k-1}}\right) = \boldsymbol{f}(\bar{\boldsymbol{X}}_{\boldsymbol{k-1}}) + \boldsymbol{w} \tag{76}$$

The Process Noise vector *w* is defined as shown in Equation 77.

$$\boldsymbol{w} = \begin{bmatrix} \boldsymbol{\sigma_{X_x}} \\ \boldsymbol{\sigma_{X_y}} \\ \boldsymbol{\sigma_{X_z}} \\ \boldsymbol{\sigma_{V_x}} \\ \boldsymbol{\sigma_{V_y}} \\ \boldsymbol{\sigma_{V_z}} \\ \boldsymbol{\sigma_{X_x}} \\ \boldsymbol{\sigma_{X_y}} \\ \boldsymbol{\sigma_{X_z}} \end{bmatrix} \tag{77}$$

The Covariance, Q, is defined as the Expected Value of the Process Noise vector *w* multiplied by the transpose of the Process Noise vector *w* as shown in Equation 78.

$$\boldsymbol{Q} = \mathbf{E}\{\boldsymbol{w}\boldsymbol{w}^{\boldsymbol{T}}\} \tag{78}$$

In order to calculate the residual required for the update step of the filter, the state estimate $\hat{X}$ must be mapped to measurement space. The mapping from state space to measurement space is achieved by using the measurement mapping equation as shown in Equation 79. Alternatively, the predicted measurement z is equivalent to the mapping of the true state vector $\bar{X}$ to measurement space plus the measurement noise vector *v*.

$$\boldsymbol{z_{k|k-1}} = \boldsymbol{h}\left(\hat{\boldsymbol{X}}_{\boldsymbol{k|k-1}}\right) = \boldsymbol{h}\left(\bar{\boldsymbol{X}}_{\boldsymbol{k|k-1}}\right) + \boldsymbol{v} \tag{79}$$

The measurement noise vector *v* is defined as shown in Equation 80

$$\boldsymbol{v} = \begin{bmatrix} \boldsymbol{\sigma_{\psi}} \\ \boldsymbol{\sigma_{\theta}} \end{bmatrix} \quad (80)$$

The Measurement Covariance, *R*, is defined as the Expected Value of the measurement noise vector *v* multiplied by the transpose of the measurement noise vector *v* as shown in Equation 81.

$$\boldsymbol{R} = \mathbf{E}\{\boldsymbol{v}\boldsymbol{v}^{\boldsymbol{T}}\} \quad (81)$$

As indicated earlier, the predicted state estimate for the next time step can be computed by applying the state transition function to the current state estimate $\hat{X}$. If the state transition function is linear, it can be replaced with a matrix multiplication as shown in Equation 82.

$$\begin{bmatrix} \hat{\boldsymbol{X}}^{\boldsymbol{R}}_{\boldsymbol{RA}_{\boldsymbol{k}}} \\ \hat{\boldsymbol{V}}^{\boldsymbol{R}}_{\boldsymbol{RA}_{\boldsymbol{k}}} \\ \hat{\boldsymbol{X}}^{\boldsymbol{R}}_{\boldsymbol{AB}_{\boldsymbol{k}}} \end{bmatrix} = \boldsymbol{F} \begin{bmatrix} \hat{\boldsymbol{X}}^{\boldsymbol{R}}_{\boldsymbol{RA}_{\boldsymbol{k-1}}} \\ \hat{\boldsymbol{V}}^{\boldsymbol{R}}_{\boldsymbol{RA}_{\boldsymbol{k-1}}} \\ \hat{\boldsymbol{X}}^{\boldsymbol{R}}_{\boldsymbol{AB}_{\boldsymbol{k-1}}} \end{bmatrix} \quad (82)$$

The matrix *F* is computed by deriving the Taylor Series approximation of the state transition functions as show in Equation 83 through Equation 85.

$$\boldsymbol{F} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{83}$$

$$\boldsymbol{\Phi} = \mathbf{I} + \mathbf{F} * \boldsymbol{\Delta t} \tag{84}$$

$$\boldsymbol{\Phi} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Delta t} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Delta t} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Delta t} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{85}$$

The measurement vector is show in Equation 86.

$$\bar{\boldsymbol{z}} = \begin{bmatrix} \boldsymbol{\psi}_{DA}^{R} \\ \boldsymbol{\theta}_{DA}^{R} \\ \boldsymbol{\psi}_{DB}^{R} \\ \boldsymbol{\theta}_{DB}^{R} \end{bmatrix} \tag{86}$$

In order to correctly apply the residual differences to the state estimate and covariance, the measurement mapping matrix must be computed. The measurement mapping matrix quantifies how errors in the measurement vector propagate to errors in the

state estimate and state covariance. The measurement mapping matrix is computed by perturbing the measurement vector with respect to the state vector. Consequently, the relationship between changes in the state vector and changes in the measurement vector is quantified in Equation 87.

$$\boldsymbol{\delta\bar{z}} = \frac{\boldsymbol{\delta z}}{\boldsymbol{\delta\bar{X}}} * \boldsymbol{\delta\bar{X}} + \boldsymbol{v} \tag{87}$$

To linearize the relationship, the Jacobian of the measurement vector is calculated with respect to the state vector, which results in the measurement mapping matrix shown in Equation 89.

$$\boldsymbol{\delta\bar{z}} = \boldsymbol{H} * \boldsymbol{\delta\bar{X}} + \boldsymbol{v} \tag{88}$$

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H_{11}} & \boldsymbol{H_{12}} & \boldsymbol{H_{13}} \\ \boldsymbol{H_{21}} & \boldsymbol{H_{22}} & \boldsymbol{H_{23}} \\ \boldsymbol{H_{31}} & \boldsymbol{H_{32}} & \boldsymbol{H_{33}} \end{bmatrix} \tag{89}$$

Equations 90 through 137 articulate the calculations required to fully specify the *H* matrix.

$$\boldsymbol{H_{11}} = \begin{bmatrix} \frac{\boldsymbol{\delta\psi_{DA}^{R}}}{\boldsymbol{\delta\vec{X}_{RA_X}^{R}}} & \frac{\boldsymbol{\delta\psi_{DA}^{R}}}{\boldsymbol{\delta\vec{X}_{RA_Y}^{R}}} & \frac{\boldsymbol{\delta\psi_{DA}^{R}}}{\boldsymbol{\delta\vec{X}_{RA_Z}^{R}}} \\ \frac{\boldsymbol{\delta\theta_{DA}^{R}}}{\boldsymbol{\delta\vec{X}_{RA_X}^{R}}} & \frac{\boldsymbol{\delta\theta_{DA}^{R}}}{\boldsymbol{\delta\vec{X}_{RA_Y}^{R}}} & \frac{\boldsymbol{\delta\theta_{DA}^{R}}}{\boldsymbol{\delta\vec{X}_{RA_Z}^{R}}} \end{bmatrix} \tag{90}$$

$$\boldsymbol{H_{12}} = \begin{bmatrix} \frac{\delta \psi_{DA}^{R}}{\delta \vec{V}_{RA_X}^{R}} & \frac{\delta \psi_{DA}^{R}}{\delta \vec{V}_{RA_Y}^{R}} & \frac{\delta \psi_{DA}^{R}}{\delta \vec{V}_{RA_Z}^{R}} \\ \frac{\delta \theta_{DA}^{R}}{\delta \vec{V}_{RA_X}^{R}} & \frac{\delta \theta_{DA}^{R}}{\delta \vec{V}_{RA_Y}^{R}} & \frac{\delta \theta_{DA}^{R}}{\delta \vec{V}_{RA_Z}^{R}} \end{bmatrix} \quad (91)$$

$$\boldsymbol{H_{13}} = \begin{bmatrix} \frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{AB_X}^{R}} & \frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{AB_Y}^{R}} & \frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{AB_Z}^{R}} \\ \frac{\delta \theta_{DA}^{R}}{\delta \vec{X}_{AB_X}^{R}} & \frac{\delta \theta_{DA}^{R}}{\delta \vec{X}_{AB_Y}^{R}} & \frac{\delta \theta_{DA}^{R}}{\delta \vec{X}_{AB_Z}^{R}} \end{bmatrix} \quad (92)$$

$$\boldsymbol{\psi_{DA}^{R}} = \mathbf{tan^{-1}}\left(\frac{\vec{X}_{DA_Y}^{R}}{\vec{X}_{DA_X}^{R}}\right) \quad (93)$$

Partial derivative of ψ_{DA}^{R} with respect to $\vec{X}_{RA}^{R}$:

$$\frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{RA_X}^{R}} = \frac{-\vec{X}_{DA_Y}^{R}}{\left(\vec{X}_{DA_X}^{R}\right)^2 + \left(\vec{X}_{DA_Y}^{R}\right)^2} \quad (94)$$

$$\frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{RA_Y}^{R}} = \frac{\vec{X}_{DA_X}^{R}}{\left(\vec{X}_{DA_X}^{R}\right)^2 + \left(\vec{X}_{DA_Y}^{R}\right)^2} \quad (95)$$

$$\frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{RA_Z}^{R}} = \mathbf{0} \quad (96)$$

Partial derivative of ψ_{DA}^{R} with respect to $\vec{V}_{RA}^{R}$:

$$\frac{\delta \psi_{DA}^{R}}{\delta \vec{V}_{RA_X}^{R}} = \frac{\delta \psi_{DA}^{R}}{\delta \vec{V}_{RA_Y}^{R}} = \frac{\delta \psi_{DA}^{R}}{\delta \vec{V}_{RA_Z}^{R}} = \mathbf{0} \quad (97)$$

Partial derivative of ψ_{DA}^{R} with respect to $\vec{X}_{AB}^{R}$:

$$\frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{AB_X}^{R}} = \frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{AB_Y}^{R}} = \frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{AB_Z}^{R}} = \mathbf{0} \tag{98}$$

Calculate the magnitude of r_{DA}:

$$\boldsymbol{r_{DA}} = \sqrt{\left(\vec{\boldsymbol{X}}_{\boldsymbol{DA_X}}^{\boldsymbol{R}}\right)^2 + \left(\vec{\boldsymbol{X}}_{\boldsymbol{DA_Y}}^{\boldsymbol{R}}\right)^2 + \left(\vec{\boldsymbol{X}}_{\boldsymbol{DA_Z}}^{\boldsymbol{R}}\right)^2} \tag{99}$$

Partial derivative of r_{DA} with respect to $\vec{X}_{RA}^{R}$:

$$\frac{\delta r_{DA}}{\delta \vec{X}_{RA_X}^{R}} = \frac{\vec{X}_{DA_X}^{R}}{r_{DA}} \tag{100}$$

$$\frac{\delta r_{DA}}{\delta \vec{X}_{RA_Y}^{R}} = \frac{\vec{X}_{DA_Y}^{R}}{r_{DA}} \tag{101}$$

$$\frac{\delta r_{DA}}{\delta \vec{X}_{RA_Z}^{R}} = \frac{\vec{X}_{DA_Z}^{R}}{r_{DA}} \tag{102}$$

Partial derivative of r_{DA} with respect to $\vec{V}_{RA}^{R}$:

$$\frac{\delta r_{DA}}{\delta \vec{V}_{RA_X}^{R}} = \frac{\delta r_{DA}}{\delta \vec{V}_{RA_Y}^{R}} = \frac{\delta r_{DA}}{\delta \vec{V}_{RA_Z}^{R}} = \mathbf{0} \tag{103}$$

Partial derivative of r_{DA} with respect to $\vec{X}_{AB}^{R}$:

$$\frac{\delta r_{DA}}{\delta \vec{X}_{AB_X}^{R}} = \frac{\delta r_{DA}}{\delta \vec{X}_{AB_Y}^{R}} = \frac{\delta r_{DA}}{\delta \vec{X}_{AB_Z}^{R}} = \mathbf{0} \quad (104)$$

To calculate the value of θ_{DA}^{R}:

$$\boldsymbol{\theta_{DA}^{R}} = -\mathbf{sin^{-1}}\left(\frac{\vec{X}_{DA_Z}^{R}}{r_{DA}}\right) \quad (105)$$

Partial derivative of θ_{DA}^{R} with respect to $\vec{X}_{RA}^{R}$:

$$\frac{\delta \theta_{DA}^{R}}{\delta \vec{X}_{RA_X}^{R}} = \frac{\vec{X}_{DA_Z}^{R} * \vec{X}_{DA_X}^{R}}{r_{DA}^{3}\sqrt{1-\left(\frac{\vec{X}_{DA_Z}^{R}}{r_{DA}}\right)^{2}}} \quad (106)$$

$$\frac{\delta \theta_{DA}^{R}}{\delta \vec{X}_{RA_Y}^{R}} = \frac{\vec{X}_{DA_Z}^{R} * \vec{X}_{DA_Y}^{R}}{r_{DA}^{3}\sqrt{1-\left(\frac{\vec{X}_{DA_Z}^{R}}{r_{DA}}\right)^{2}}} \quad (107)$$

$$\frac{\delta \theta_{DA}^{R}}{\delta \vec{X}_{RA_Z}^{R}} = \frac{-r_{DA}^{2}+\left(\vec{X}_{DA_Z}^{R}\right)^{2}}{r_{DA}^{3}\sqrt{1-\left(\frac{\vec{X}_{DA_Z}^{R}}{r_{DA}}\right)^{2}}} \quad (108)$$

Partial derivative of θ_{DA}^{R} with respect to $\vec{V}_{RA}^{R}$:

$$\frac{\delta\theta_{DA}^{R}}{\delta\vec{V}_{RA_X}^{R}} = \frac{\delta\theta_{DA}^{R}}{\delta\vec{V}_{DA_Y}^{R}} = \frac{\delta\theta_{DA}^{R}}{\delta\vec{V}_{DA_Z}^{R}} = \mathbf{0} \tag{109}$$

Partial derivative of θ_{DA}^{R} with respect to $\vec{X}_{AB}^{R}$:

$$\frac{\delta\theta_{DA}^{R}}{\delta\vec{X}_{AB_X}^{R}} = \frac{\delta\theta_{DA}^{R}}{\delta\vec{X}_{AB_Y}^{R}} = \frac{\delta\theta_{DA}^{R}}{\delta\vec{X}_{AB_Z}^{R}} = \mathbf{0} \tag{110}$$

Find ψ_{DB}^{R} and θ_{DB}^{R} mapping equations:

$$\boldsymbol{H_{21}} = \begin{bmatrix} \frac{\delta\psi_{DB}^{R}}{\delta\vec{X}_{RA_X}^{R}} & \frac{\delta\psi_{DB}^{R}}{\delta\vec{X}_{RA_Y}^{R}} & \frac{\delta\psi_{DB}^{R}}{\delta\vec{X}_{RA_Z}^{R}} \\ \frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{RA_X}^{R}} & \frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{RA_Y}^{R}} & \frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{RA_Z}^{R}} \end{bmatrix} \tag{111}$$

$$\boldsymbol{H_{22}} = \begin{bmatrix} \frac{\delta\psi_{DB}^{R}}{\delta\vec{V}_{RA_X}^{R}} & \frac{\delta\psi_{DB}^{R}}{\delta\vec{V}_{RA_Y}^{R}} & \frac{\delta\psi_{DB}^{R}}{\delta\vec{V}_{RA_Z}^{R}} \\ \frac{\delta\theta_{DB}^{R}}{\delta\vec{V}_{RA_X}^{R}} & \frac{\delta\theta_{DB}^{R}}{\delta\vec{V}_{RA_Y}^{R}} & \frac{\delta\theta_{DB}^{R}}{\delta\vec{V}_{RA_Z}^{R}} \end{bmatrix} \tag{112}$$

$$\boldsymbol{H_{23}} = \begin{bmatrix} \frac{\delta\psi_{DB}^{R}}{\delta\vec{X}_{AB_X}^{R}} & \frac{\delta\psi_{DB}^{R}}{\delta\vec{X}_{AB_Y}^{R}} & \frac{\delta\psi_{DB}^{R}}{\delta\vec{X}_{AB_Z}^{R}} \\ \frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{AB_X}^{R}} & \frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{AB_Y}^{R}} & \frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{AB_Z}^{R}} \end{bmatrix} \tag{113}$$

To calculate ψ_{DB}^{R}:

$$\boldsymbol{\psi_{DB}^{R} = \tan^{-1}\left(\frac{\vec{X}_{DA_Y}^{R} + \vec{X}_{AB_Y}^{R}}{\vec{X}_{DA_X}^{R} + \vec{X}_{AB_X}^{R}}\right)} \quad (114)$$

Partial derivative of ψ_{DB}^{R} with respect to $\vec{X}_{RA}^{R}$:

$$\boldsymbol{\frac{\delta\psi_{DB}^{R}}{\delta\vec{X}_{RA_X}^{R}} = \frac{-\left(\vec{X}_{DA_Y}^{R} + \vec{X}_{AB_Y}^{R}\right)}{\left(\vec{X}_{DA_X}^{R} + \vec{X}_{AB_X}^{R}\right)^2 + \left(\vec{X}_{DA_Y}^{R} + \vec{X}_{AB_Y}^{R}\right)^2}} \quad (115)$$

$$\boldsymbol{\frac{\delta\psi_{DB}^{R}}{\delta\vec{X}_{RA_Y}^{R}} = \frac{\vec{X}_{DA_X}^{R} + \vec{X}_{AB_X}^{R}}{\left(\vec{X}_{DA_X}^{R} + \vec{X}_{AB_X}^{R}\right)^2 + \left(\vec{X}_{DA_Y}^{R} + \vec{X}_{AB_Y}^{R}\right)^2}} \quad (116)$$

$$\boldsymbol{\frac{\delta\psi_{DB}^{R}}{\delta\vec{X}_{RA_Z}^{R}} = 0} \quad (117)$$

Partial derivative of ψ_{DB}^{R} with respect to $\vec{V}_{RA}^{R}$:

$$\boldsymbol{\frac{\delta\psi_{DB}^{R}}{\delta\vec{V}_{RA_X}^{R}} = \frac{\delta\psi_{DB}^{R}}{\delta\vec{V}_{RA_Y}^{R}} = \frac{\delta\psi_{DB}^{R}}{\delta\vec{V}_{RA_Z}^{R}} = 0} \quad (118)$$

Partial derivative of ψ_{DB}^{R} with respect to $\vec{X}_{AB}^{R}$:

$$\boldsymbol{\frac{\delta\psi_{DB}^{R}}{\delta\vec{X}_{AB_X}^{R}} = \frac{-\left(\vec{X}_{DA_Y}^{R} + \vec{X}_{AB_Y}^{R}\right)}{\left(\vec{X}_{DA_X}^{R} + \vec{X}_{AB_X}^{R}\right)^2 + \left(\vec{X}_{DA_Y}^{R} + \vec{X}_{AB_Y}^{R}\right)^2}} \quad (119)$$

$$\frac{\delta \psi_{DB}^{R}}{\delta \vec{X}_{AB_Y}^{R}} = \frac{\vec{X}_{DA_X}^{R} + \vec{X}_{AB_X}^{R}}{\left(\vec{X}_{DA_X}^{R} + \vec{X}_{AB_X}^{R}\right)^2 + \left(\vec{X}_{DA_Y}^{R} + \vec{X}_{AB_Y}^{R}\right)^2} \quad (120)$$

$$\frac{\delta \psi_{DB}^{R}}{\delta \vec{X}_{AB_Z}^{R}} = \mathbf{0} \quad (121)$$

To calculate the magnitude of r_{DB}:

$$\boldsymbol{r_{DB}} = \sqrt{\left(\vec{\boldsymbol{X}}_{\boldsymbol{DA_X}}^{\boldsymbol{R}} + \vec{\boldsymbol{X}}_{\boldsymbol{AB_X}}^{\boldsymbol{R}}\right)^2 + \left(\vec{\boldsymbol{X}}_{\boldsymbol{DA_Y}}^{\boldsymbol{R}} + \vec{\boldsymbol{X}}_{\boldsymbol{AB_Y}}^{\boldsymbol{R}}\right)^2 + \left(\vec{\boldsymbol{X}}_{\boldsymbol{DA_Z}}^{\boldsymbol{R}} + \vec{\boldsymbol{X}}_{\boldsymbol{AB_Z}}^{\boldsymbol{R}}\right)^2} \quad (122)$$

Partial derivative of r_{DB} with respect to $\vec{X}_{RA}^{R}$:

$$\frac{\delta r_{DB}}{\delta \vec{X}_{RA_X}^{R}} = \frac{\vec{X}_{DA_X}^{R} + \vec{X}_{AB_X}^{R}}{r_{DB}} \quad (123)$$

$$\frac{\delta r_{DB}}{\delta \vec{X}_{RA_Y}^{R}} = \frac{\vec{X}_{DA_Y}^{R} + \vec{X}_{AB_Y}^{R}}{r_{DB}} \quad (124)$$

$$\frac{\delta r_{DB}}{\delta \vec{X}_{RA_Z}^{R}} = \frac{\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}}{r_{DB}} \quad (125)$$

Partial derivative of r_{DB} with respect to $\vec{V}_{RA}^{R}$:

$$\frac{\delta r_{DB}}{\delta \vec{V}_{RA_X}^{R}} = \frac{\delta r_{DB}}{\delta \vec{V}_{RA_Y}^{R}} = \frac{\delta r_{DB}}{\delta \vec{V}_{RA_Z}^{R}} = \mathbf{0} \quad (126)$$

Partial derivative of r_{DB} with respect to $\vec{X}_{AB}^{R}$:

$$\frac{\delta r_{DB}}{\delta \vec{X}_{AB_X}^{R}} = \frac{\vec{X}_{DA_X}^{R} + \vec{X}_{AB_X}^{R}}{r_{DB}} \quad (127)$$

$$\frac{\delta r_{DB}}{\delta \vec{X}_{AB_Y}^{R}} = \frac{\vec{X}_{DA_Y}^{R} + \vec{X}_{AB_Y}^{R}}{r_{DB}} \quad (128)$$

$$\frac{\delta r_{DB}}{\delta \vec{X}_{AB_Z}^{R}} = \frac{\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}}{r_{DB}} \quad (129)$$

To calculate the value of θ_{DB}^{R}:

$$\boldsymbol{\theta_{DB}^{R}} = -\mathbf{sin^{-1}}\left(\frac{\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}}{r_{DB}}\right) \quad (130)$$

Partial derivative of θ_{DB}^{R} with respect to $\vec{X}_{RA}^{R}$:

$$\frac{\delta \theta_{DB}^{R}}{\delta \vec{X}_{RA_X}^{R}} = \frac{\left(\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}\right) * \left(\vec{X}_{DA_X}^{R} + \vec{X}_{AB_X}^{R}\right)}{r_{DB}^{3}\sqrt{1 - \left(\frac{\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}}{r_{DB}}\right)^{2}}} \quad (131)$$

$$\frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{RA_Y}^{R}} = \frac{\left(\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}\right)*\left(\vec{X}_{DA_Y}^{R} + \vec{X}_{AB_Y}^{R}\right)}{r_{DB}^{3}\sqrt{1-\left(\frac{\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}}{r_{DB}}\right)^{2}}} \quad (132)$$

$$\frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{RA_Z}^{R}} = \frac{-r_{DB}^{2}+\left(\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}\right)^{2}}{r_{DB}^{3}\sqrt{1-\left(\frac{\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}}{r_{DB}}\right)^{2}}} \quad (133)$$

Partial derivative of θ_{DB}^{R} with respect to $\vec{V}_{RA}^{R}$:

$$\frac{\delta\theta_{DB}^{R}}{\delta\vec{V}_{RA_X}^{R}} = \frac{\delta\theta_{DB}^{R}}{\delta\vec{V}_{RA_Y}^{R}} = \frac{\delta\theta_{DB}^{R}}{\delta\vec{V}_{RA_Z}^{R}} = \mathbf{0} \quad (134)$$

Partial derivative of θ_{DB}^{R} with respect to $\vec{X}_{AB}^{R}$:

$$\frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{AB_X}^{R}} = \frac{\left(\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}\right)*\left(\vec{X}_{DA_X}^{R} + \vec{X}_{AB_X}^{R}\right)}{r_{DB}^{3}\sqrt{1-\left(\frac{\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}}{r_{DB}}\right)^{2}}} \quad (135)$$

$$\frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{AB_Y}^{R}} = \frac{\left(\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}\right)*\left(\vec{X}_{DA_Y}^{R} + \vec{X}_{AB_Y}^{R}\right)}{r_{DB}^{3}\sqrt{1-\left(\frac{\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}}{r_{DB}}\right)^{2}}} \quad (136)$$

$$\frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{AB_Z}^{R}} = \frac{-r_{DB}^{2}+\left(\vec{X}_{DA_Z}^{R}+\vec{X}_{AB_Z}^{R}\right)^{2}}{r_{DB}^{3}\sqrt{1-\left(\frac{\vec{X}_{DA_Z}^{R}+\vec{X}_{AB_Z}^{R}}{r_{DB}}\right)^{2}}} \quad (137)$$

Twelve State Filter Derivation

Use of a multiple six state track filter approach requires that each PoI be tracked independently of one another. Such an approach does not utilize the known geometric relationship between the points. By tracking the three point geometry in a single filter, the a priori knowledge of the rigid geometry can be used to improve the state estimate and the classification results. The first step of the derivation of the Twelve State filter is to define the filter state as shown in Equation 138.

$$\widehat{\boldsymbol{X}} = \begin{bmatrix} \widehat{\boldsymbol{X}}_{RA}^{R} \\ \widehat{\boldsymbol{V}}_{RA}^{R} \\ \widehat{\boldsymbol{X}}_{AB}^{R} \\ \widehat{\boldsymbol{X}}_{AC}^{R} \end{bmatrix} \quad (138)$$

The predicted state estimate for the next time step can be computed by applying the state transition function to the current state estimate $\hat{X}$. Alternatively, the predicted state is equivalent to the state transition function applied to the true state $\bar{X}$ plus the Process Noise vector *w*, as shown in Equation 139.

$$\widehat{\boldsymbol{X}}_{\boldsymbol{k|k-1}} = \boldsymbol{f}\left(\widehat{\boldsymbol{X}}_{\boldsymbol{k-1}}\right) = \boldsymbol{f}(\overline{\boldsymbol{X}}_{\boldsymbol{k-1}}) + \boldsymbol{w} \quad (139)$$

The Process Noise vector *w* is defined as shown in Equation 140.

$$\boldsymbol{w} = \begin{bmatrix} \boldsymbol{\sigma_{X_x}} \\ \boldsymbol{\sigma_{X_y}} \\ \boldsymbol{\sigma_{X_z}} \\ \boldsymbol{\sigma_{V_x}} \\ \boldsymbol{\sigma_{V_y}} \\ \boldsymbol{\sigma_{V_z}} \\ \boldsymbol{\sigma_{X_x}} \\ \boldsymbol{\sigma_{X_y}} \\ \boldsymbol{\sigma_{X_z}} \\ \boldsymbol{\sigma_{X_x}} \\ \boldsymbol{\sigma_{X_y}} \\ \boldsymbol{\sigma_{X_z}} \end{bmatrix} \tag{140}$$

The Covariance, Q, is defined as the Expected Value of the Process Noise vector *w* multiplied by the transpose of the Process Noise vector *w* as shown in Equation 141.

$$\boldsymbol{Q} = \mathbf{E}\{\boldsymbol{w}\boldsymbol{w}^{\boldsymbol{T}}\} \tag{141}$$

In order to calculate the residual required for the update step of the filter, the state estimate $\hat{X}$ must be mapped to measurement space. The mapping from state space to measurement space is achieved by using the measurement mapping equation as shown in Equation 142. Alternatively, the predicted measurement z is equivalent to the mapping of the true state vector $\bar{X}$ to measurement space plus the measurement noise vector *v*.

$$\boldsymbol{z_{k|k-1}} = \boldsymbol{h}\left(\hat{\boldsymbol{X}}_{\boldsymbol{k|k-1}}\right) = \boldsymbol{h}\left(\bar{\boldsymbol{X}}_{\boldsymbol{k|k-1}}\right) + \boldsymbol{v} \tag{142}$$

The measurement noise vector *v* is defined as shown in Equation 143

$$\boldsymbol{v} = \begin{bmatrix} \boldsymbol{\sigma_{\psi}} \\ \boldsymbol{\sigma_{\theta}} \\ \boldsymbol{\sigma_{\psi}} \\ \boldsymbol{\sigma_{\theta}} \\ \boldsymbol{\sigma_{\psi}} \\ \boldsymbol{\sigma_{\theta}} \end{bmatrix} \tag{143}$$

The Measurement Covariance, *R*, is defined as the Expected Value of the measurement noise vector *v* multiplied by the transpose of the measurement noise vector *v* as shown in Equation 144.

$$\boldsymbol{R} = \mathbf{E}\{\boldsymbol{v}\boldsymbol{v}^{\boldsymbol{T}}\} \tag{144}$$

As indicated earlier, the predicted state estimate for the next time step can be computed by applying the state transition function to the current state estimate $\hat{\boldsymbol{X}}$. If the state transition function is linear, it can be replaced with a matrix multiplication as shown in Equation 145

$$\begin{bmatrix} \widehat{\boldsymbol{X}}^{\boldsymbol{R}}_{\boldsymbol{RA}_{\boldsymbol{k}}} \\ \widehat{\boldsymbol{V}}^{\boldsymbol{R}}_{\boldsymbol{RA}_{\boldsymbol{k}}} \\ \widehat{\boldsymbol{X}}^{\boldsymbol{R}}_{\boldsymbol{AB}_{\boldsymbol{k}}} \\ \widehat{\boldsymbol{X}}^{\boldsymbol{R}}_{\boldsymbol{AC}_{\boldsymbol{k}}} \end{bmatrix} = \boldsymbol{F} \begin{bmatrix} \widehat{\boldsymbol{X}}^{\boldsymbol{R}}_{\boldsymbol{RA}_{\boldsymbol{k-1}}} \\ \widehat{\boldsymbol{V}}^{\boldsymbol{R}}_{\boldsymbol{RA}_{\boldsymbol{k-1}}} \\ \widehat{\boldsymbol{X}}^{\boldsymbol{R}}_{\boldsymbol{AB}_{\boldsymbol{k-1}}} \\ \widehat{\boldsymbol{X}}^{\boldsymbol{R}}_{\boldsymbol{AC}_{\boldsymbol{k-1}}} \end{bmatrix} \tag{145}$$

The matrix *F* is computed by deriving the Taylor Series approximation of the state transition functions as show in Equation 146 through Equation 148.

$$\boldsymbol{F} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (146)$$

$$\boldsymbol{\Phi} = \mathbf{I} + \mathbf{F} * \boldsymbol{\Delta t} \quad (147)$$

$$\boldsymbol{\Phi} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Delta t} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Delta t} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Delta t} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \quad (148)$$

The measurement vector is show in Equation 149.

$$\bar{\boldsymbol{z}} = \begin{bmatrix} \boldsymbol{\psi}_{DA}^{R} \\ \boldsymbol{\theta}_{DA}^{R} \\ \boldsymbol{\psi}_{DB}^{R} \\ \boldsymbol{\theta}_{DB}^{R} \\ \boldsymbol{\psi}_{DC}^{R} \\ \boldsymbol{\theta}_{DC}^{R} \end{bmatrix} \quad (149)$$

In order to correctly apply the residual differences to the state estimate and covariance, the measurement mapping matrix must be computed. The measurement mapping matrix quantifies how errors in the measurement vector propagate to errors in the state estimate and state covariance. The measurement mapping matrix is computed by perturbing the measurement vector with respect to the state vector. Consequently, the relationship between changes in the state vector and changes in the measurement vector is quantified in Equation 150.

$$\boldsymbol{\delta\bar{z}} = \frac{\boldsymbol{\delta z}}{\boldsymbol{\delta\bar{X}}} * \boldsymbol{\delta\bar{X}} + \boldsymbol{v} \tag{150}$$

To linearize the relationship, the Jacobian of the measurement vector is calculated with respect to the state vector, which results in the measurement mapping matrix shown in Equation 152.

$$\boldsymbol{\delta\bar{z}} = \boldsymbol{H} * \boldsymbol{\delta\bar{X}} + \boldsymbol{v} \tag{151}$$

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H_{11}} & \boldsymbol{H_{12}} & \boldsymbol{H_{13}} & \boldsymbol{H_{14}} \\ \boldsymbol{H_{21}} & \boldsymbol{H_{22}} & \boldsymbol{H_{23}} & \boldsymbol{H_{24}} \\ \boldsymbol{H_{31}} & \boldsymbol{H_{32}} & \boldsymbol{H_{33}} & \boldsymbol{H_{34}} \end{bmatrix} \tag{152}$$

Equations 153 through 239 articulate the calculations required to fully specify the *H* matrix.

$$\boldsymbol{H_{11}} = \begin{bmatrix} \frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{RA_X}^{R}} & \frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{RA_Y}^{R}} & \frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{RA_Z}^{R}} \\ \frac{\delta \theta_{DA}^{R}}{\delta \vec{X}_{RA_X}^{R}} & \frac{\delta \theta_{DA}^{R}}{\delta \vec{X}_{RA_Y}^{R}} & \frac{\delta \theta_{DA}^{R}}{\delta \vec{X}_{RA_Z}^{R}} \end{bmatrix} \quad (153)$$

$$\boldsymbol{H_{12}} = \begin{bmatrix} \frac{\delta \psi_{DA}^{R}}{\delta \vec{V}_{RA_X}^{R}} & \frac{\delta \psi_{DA}^{R}}{\delta \vec{V}_{RA_Y}^{R}} & \frac{\delta \psi_{DA}^{R}}{\delta \vec{V}_{RA_Z}^{R}} \\ \frac{\delta \theta_{DA}^{R}}{\delta \vec{V}_{RA_X}^{R}} & \frac{\delta \theta_{DA}^{R}}{\delta \vec{V}_{RA_Y}^{R}} & \frac{\delta \theta_{DA}^{R}}{\delta \vec{V}_{RA_Z}^{R}} \end{bmatrix} \quad (154)$$

$$\boldsymbol{H_{13}} = \begin{bmatrix} \frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{AB_X}^{R}} & \frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{AB_Y}^{R}} & \frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{AB_Z}^{R}} \\ \frac{\delta \theta_{DA}^{R}}{\delta \vec{X}_{AB_X}^{R}} & \frac{\delta \theta_{DA}^{R}}{\delta \vec{X}_{AB_Y}^{R}} & \frac{\delta \theta_{DA}^{R}}{\delta \vec{X}_{AB_Z}^{R}} \end{bmatrix} \quad (155)$$

$$\boldsymbol{H_{14}} = \begin{bmatrix} \frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{AC_X}^{R}} & \frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{AC_Y}^{R}} & \frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{AC_Z}^{R}} \\ \frac{\delta \theta_{DA}^{R}}{\delta \vec{X}_{AC_X}^{R}} & \frac{\delta \theta_{DA}^{R}}{\delta \vec{X}_{AC_Y}^{R}} & \frac{\delta \theta_{DA}^{R}}{\delta \vec{X}_{AC_Z}^{R}} \end{bmatrix} \quad (156)$$

To calculate ψ_{DA}^{R}:

$$\boldsymbol{\psi_{DA}^{R}} = \mathbf{tan^{-1}}\left(\frac{\vec{X}_{DA_Y}^{R}}{\vec{X}_{DA_X}^{R}}\right) \quad (157)$$

Partial derivative of ψ_{DA}^{R} with respect to $\vec{X}_{RA}^{R}$:

$$\frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{RA_X}^{R}} = \frac{-\vec{X}_{DA_Y}^{R}}{\left(\vec{X}_{DA_X}^{R}\right)^2 + \left(\vec{X}_{DA_Y}^{R}\right)^2} \quad (158)$$

$$\frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{RA_Y}^{R}} = \frac{\vec{X}_{DA_X}^{R}}{\left(\vec{X}_{DA_X}^{R}\right)^2 + \left(\vec{X}_{DA_Y}^{R}\right)^2} \quad (159)$$

$$\frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{RA_Z}^{R}} = \mathbf{0} \quad (160)$$

Partial derivative of ψ_{DA}^{R} with respect to $\vec{V}_{RA}^{R}$:

$$\frac{\delta \psi_{DA}^{R}}{\delta \vec{V}_{RA_X}^{R}} = \frac{\delta \psi_{DA}^{R}}{\delta \vec{V}_{RA_Y}^{R}} = \frac{\delta \psi_{DA}^{R}}{\delta \vec{V}_{RA_Z}^{R}} = \mathbf{0} \quad (161)$$

Partial derivative of ψ_{DA}^{R} with respect to $\vec{X}_{AB}^{R}$:

$$\frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{AB_X}^{R}} = \frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{AB_Y}^{R}} = \frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{AB_Z}^{R}} = \mathbf{0} \quad (162)$$

Partial derivative of ψ_{DA}^{R} with respect to $\vec{X}_{AC}^{R}$:

$$\frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{AC_X}^{R}} = \frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{AC_Y}^{R}} = \frac{\delta \psi_{DA}^{R}}{\delta \vec{X}_{AC_Z}^{R}} = \mathbf{0} \quad (163)$$

To calculate the magnitude of vector r_{DA}:

$$\boldsymbol{r_{DA}} = \sqrt{\left(\vec{\boldsymbol{X}}_{\boldsymbol{DA_X}}^{\boldsymbol{R}}\right)^2 + \left(\vec{\boldsymbol{X}}_{\boldsymbol{DA_Y}}^{\boldsymbol{R}}\right)^2 + \left(\vec{\boldsymbol{X}}_{\boldsymbol{DA_Z}}^{\boldsymbol{R}}\right)^2} \quad (164)$$

Partial derivative of r_{DA} with respect to $\vec{X}_{RA}^{R}$:

$$\frac{\delta r_{DA}}{\delta \vec{X}_{RA_X}^{R}} = \frac{\vec{X}_{DA_X}^{R}}{r_{DA}} \quad (165)$$

$$\frac{\delta r_{DA}}{\delta \vec{X}_{RA_Y}^{R}} = \frac{\vec{X}_{DA_Y}^{R}}{r_{DA}} \quad (166)$$

$$\frac{\delta r_{DA}}{\delta \vec{X}_{RA_Z}^{R}} = \frac{\vec{X}_{DA_Z}^{R}}{r_{DA}} \quad (167)$$

Partial derivative of r_{DA} with respect to $\vec{V}_{RA}^{R}$:

$$\frac{\delta r_{DA}}{\delta \vec{V}_{RA_X}^{R}} = \frac{\delta r_{DA}}{\delta \vec{V}_{RA_Y}^{R}} = \frac{\delta r_{DA}}{\delta \vec{V}_{RA_Z}^{R}} = \mathbf{0} \quad (168)$$

Partial derivative of r_{DA} with respect to $\vec{X}_{AB}^{R}$:

$$\frac{\delta r_{DA}}{\delta \vec{X}_{AB_X}^{R}} = \frac{\delta r_{DA}}{\delta \vec{X}_{AB_Y}^{R}} = \frac{\delta r_{DA}}{\delta \vec{X}_{AB_Z}^{R}} = \mathbf{0} \quad (169)$$

Partial derivative of r_{DA} with respect to $\vec{X}_{AC}^{R}$:

$$\frac{\delta r_{DA}}{\delta \vec{X}_{AC_X}^{R}} = \frac{\delta r_{DA}}{\delta \vec{X}_{AC_Y}^{R}} = \frac{\delta r_{DA}}{\delta \vec{X}_{AC_Z}^{R}} = \mathbf{0} \quad (170)$$

To calculate the value of θ_{DA}^{R}:

$$\boldsymbol{\theta_{DA}^{R} = -\sin^{-1}\left(\frac{\vec{X}_{DA_Z}^{R}}{r_{DA}}\right)} \quad (171)$$

Partial derivative of θ_{DA}^{R} with respect to $\vec{X}_{RA}^{R}$:

$$\frac{\delta\theta_{DA}^{R}}{\delta\vec{X}_{RA_X}^{R}} = \frac{\vec{X}_{DA_Z}^{R} * \vec{X}_{DA_X}^{R}}{r_{DA}^{3}\sqrt{1-\left(\frac{\vec{X}_{DA_Z}^{R}}{r_{DA}}\right)^{2}}} \quad (172)$$

$$\frac{\delta\theta_{DA}^{R}}{\delta\vec{X}_{RA_Y}^{R}} = \frac{\vec{X}_{DA_Z}^{R} * \vec{X}_{DA_Y}^{R}}{r_{DA}^{3}\sqrt{1-\left(\frac{\vec{X}_{DA_Z}^{R}}{r_{DA}}\right)^{2}}} \quad (173)$$

$$\frac{\delta\theta_{DA}^{R}}{\delta\vec{X}_{RA_Z}^{R}} = \frac{-r_{DA}^{2}+\left(\vec{X}_{DA_Z}^{R}\right)^{2}}{r_{DA}^{3}\sqrt{1-\left(\frac{\vec{X}_{DA_Z}^{R}}{r_{DA}}\right)^{2}}} \quad (174)$$

Partial derivative of θ_{DA}^{R} with respect to $\vec{V}_{RA}^{R}$:

$$\frac{\delta\theta_{DA}^{R}}{\delta\vec{V}_{RA_X}^{R}} = \frac{\delta\theta_{DA}^{R}}{\delta\vec{V}_{DA_Y}^{R}} = \frac{\delta\theta_{DA}^{R}}{\delta\vec{V}_{DA_Z}^{R}} = \mathbf{0} \quad (175)$$

Partial derivative of θ_{DA}^{R} with respect to $\vec{X}_{AB}^{R}$:

$$\frac{\delta\theta_{DA}^{R}}{\delta\vec{X}_{AB_X}^{R}} = \frac{\delta\theta_{DA}^{R}}{\delta\vec{X}_{AB_Y}^{R}} = \frac{\delta\theta_{DA}^{R}}{\delta\vec{X}_{AB_Z}^{R}} = \mathbf{0} \tag{176}$$

Partial derivative of θ_{DA}^{R} with respect to $\vec{X}_{AC}^{R}$:

$$\frac{\delta\theta_{DA}^{R}}{\delta\vec{X}_{AC_X}^{R}} = \frac{\delta\theta_{DA}^{R}}{\delta\vec{X}_{AC_Y}^{R}} = \frac{\delta\theta_{DA}^{R}}{\delta\vec{X}_{AC_Z}^{R}} = \mathbf{0} \tag{177}$$

Find ψ_{DB}^{R} and θ_{DB}^{R} mapping equations:

$$\boldsymbol{H_{21}} = \begin{bmatrix} \frac{\delta\psi_{DB}^{R}}{\delta\vec{X}_{RA_X}^{R}} & \frac{\delta\psi_{DB}^{R}}{\delta\vec{X}_{RA_Y}^{R}} & \frac{\delta\psi_{DB}^{R}}{\delta\vec{X}_{RA_Z}^{R}} \\ \frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{RA_X}^{R}} & \frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{RA_Y}^{R}} & \frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{RA_Z}^{R}} \end{bmatrix} \tag{178}$$

$$\boldsymbol{H_{22}} = \begin{bmatrix} \frac{\delta\psi_{DB}^{R}}{\delta\vec{V}_{RA_X}^{R}} & \frac{\delta\psi_{DB}^{R}}{\delta\vec{V}_{RA_Y}^{R}} & \frac{\delta\psi_{DB}^{R}}{\delta\vec{V}_{RA_Z}^{R}} \\ \frac{\delta\theta_{DB}^{R}}{\delta\vec{V}_{RA_X}^{R}} & \frac{\delta\theta_{DB}^{R}}{\delta\vec{V}_{RA_Y}^{R}} & \frac{\delta\theta_{DB}^{R}}{\delta\vec{V}_{RA_Z}^{R}} \end{bmatrix} \tag{179}$$

$$\boldsymbol{H_{23}} = \begin{bmatrix} \frac{\delta\psi_{DB}^{R}}{\delta\vec{X}_{AB_X}^{R}} & \frac{\delta\psi_{DB}^{R}}{\delta\vec{X}_{AB_Y}^{R}} & \frac{\delta\psi_{DB}^{R}}{\delta\vec{X}_{AB_Z}^{R}} \\ \frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{AB_X}^{R}} & \frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{AB_Y}^{R}} & \frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{AB_Z}^{R}} \end{bmatrix} \tag{180}$$

$$\boldsymbol{H_{24}} = \begin{bmatrix} \frac{\delta \psi_{DB}^{R}}{\delta \vec{X}_{AC_X}^{R}} & \frac{\delta \psi_{DB}^{R}}{\delta \vec{X}_{AC_Y}^{R}} & \frac{\delta \psi_{DB}^{R}}{\delta \vec{X}_{AC_z}^{R}} \\ \frac{\delta \theta_{DB}^{R}}{\delta \vec{X}_{AC_X}^{R}} & \frac{\delta \theta_{DB}^{R}}{\delta \vec{X}_{AC_Y}^{R}} & \frac{\delta \theta_{DB}^{R}}{\delta \vec{X}_{AC_z}^{R}} \end{bmatrix} \quad (181)$$

To calculate ψ_{DB}^{R}:

$$\boldsymbol{\psi_{DB}^{R}} = \mathbf{tan^{-1}}\left(\frac{\vec{X}_{DA_Y}^{R} + \vec{X}_{AB_Y}^{R}}{\vec{X}_{DA_X}^{R} + \vec{X}_{AB_X}^{R}}\right) \quad (182)$$

Partial derivative of ψ_{DB}^{R} with respect to $\vec{X}_{RA}^{R}$:

$$\frac{\delta \psi_{DB}^{R}}{\delta \vec{X}_{RA_X}^{R}} = \frac{-\left(\vec{X}_{DA_Y}^{R} + \vec{X}_{AB_Y}^{R}\right)}{\left(\vec{X}_{DA_X}^{R} + \vec{X}_{AB_X}^{R}\right)^2 + \left(\vec{X}_{DA_Y}^{R} + \vec{X}_{AB_Y}^{R}\right)^2} \quad (183)$$

$$\frac{\delta \psi_{DB}^{R}}{\delta \vec{X}_{RA_Y}^{R}} = \frac{\vec{X}_{DA_X}^{R} + \vec{X}_{AB_X}^{R}}{\left(\vec{X}_{DA_X}^{R} + \vec{X}_{AB_X}^{R}\right)^2 + \left(\vec{X}_{DA_Y}^{R} + \vec{X}_{AB_Y}^{R}\right)^2} \quad (184)$$

$$\frac{\delta \psi_{DB}^{R}}{\delta \vec{X}_{RA_z}^{R}} = \mathbf{0} \quad (185)$$

Partial derivative of ψ_{DB}^{R} with respect to $\vec{V}_{RA}^{R}$:

$$\frac{\delta \psi_{DB}^{R}}{\delta \vec{V}_{RA_X}^{R}} = \frac{\delta \psi_{DB}^{R}}{\delta \vec{V}_{RA_Y}^{R}} = \frac{\delta \psi_{DB}^{R}}{\delta \vec{V}_{RA_Z}^{R}} = \mathbf{0} \quad (186)$$

Partial derivative of ψ_{DB}^{R} with respect to $\vec{X}_{AB}^{R}$:

$$\frac{\delta\psi_{DB}^{R}}{\delta\vec{X}_{AB_X}^{R}} = \frac{-\left(\vec{X}_{DA_Y}^{R} + \vec{X}_{AB_Y}^{R}\right)}{\left(\vec{X}_{DA_X}^{R} + \vec{X}_{AB_X}^{R}\right)^2 + \left(\vec{X}_{DA_Y}^{R} + \vec{X}_{AB_Y}^{R}\right)^2} \quad (187)$$

$$\frac{\delta\psi_{DB}^{R}}{\delta\vec{X}_{AB_Y}^{R}} = \frac{\vec{X}_{DA_X}^{R} + \vec{X}_{AB_X}^{R}}{\left(\vec{X}_{DA_X}^{R} + \vec{X}_{AB_X}^{R}\right)^2 + \left(\vec{X}_{DA_Y}^{R} + \vec{X}_{AB_Y}^{R}\right)^2} \quad (188)$$

$$\frac{\delta\psi_{DB}^{R}}{\delta\vec{X}_{AB_Z}^{R}} = \mathbf{0} \quad (189)$$

Partial derivative of ψ_{DB}^{R} with respect to $\vec{X}_{AC}^{R}$:

$$\frac{\delta\psi_{DB}^{R}}{\delta\vec{X}_{AC_X}^{R}} = \frac{\delta\psi_{DB}^{R}}{\delta\vec{X}_{AC_Y}^{R}} = \frac{\delta\psi_{DB}^{R}}{\delta\vec{X}_{AC_Z}^{R}} = \mathbf{0} \quad (190)$$

To calculate the magnitude of r_{DB}:

$$\boldsymbol{r_{DB}} = \sqrt{\left(\vec{\boldsymbol{X}}_{\boldsymbol{DA_X}}^{\boldsymbol{R}} + \vec{\boldsymbol{X}}_{\boldsymbol{AB_X}}^{\boldsymbol{R}}\right)^2 + \left(\vec{\boldsymbol{X}}_{\boldsymbol{DA_Y}}^{\boldsymbol{R}} + \vec{\boldsymbol{X}}_{\boldsymbol{AB_Y}}^{\boldsymbol{R}}\right)^2 + \left(\vec{\boldsymbol{X}}_{\boldsymbol{DA_Z}}^{\boldsymbol{R}} + \vec{\boldsymbol{X}}_{\boldsymbol{AB_Z}}^{\boldsymbol{R}}\right)^2} \quad (191)$$

Partial derivative of r_{DB} with respect to $\vec{X}_{RA}^{R}$:

$$\frac{\delta r_{DB}}{\delta\vec{X}_{RA_X}^{R}} = \frac{\vec{X}_{DA_X}^{R} + \vec{X}_{AB_X}^{R}}{r_{DB}} \quad (192)$$

$$\frac{\delta r_{DB}}{\delta \vec{X}^R_{RA_Y}} = \frac{\vec{X}^R_{DA_Y} + \vec{X}^R_{AB_Y}}{r_{DB}} \quad (193)$$

$$\frac{\delta r_{DB}}{\delta \vec{X}^R_{RA_Z}} = \frac{\vec{X}^R_{DA_Z} + \vec{X}^R_{AB_Z}}{r_{DB}} \quad (194)$$

Partial derivative of r_{DB} with respect to $\vec{V}^R_{RA}$:

$$\frac{\delta r_{DB}}{\delta \vec{V}^R_{RA_X}} = \frac{\delta r_{DB}}{\delta \vec{V}^R_{RA_Y}} = \frac{\delta r_{DB}}{\delta \vec{V}^R_{RA_Z}} = \mathbf{0} \quad (195)$$

Partial derivative of r_{DB} with respect to $\vec{X}^R_{AB}$:

$$\frac{\delta r_{DB}}{\delta \vec{X}^R_{AB_X}} = \frac{\vec{X}^R_{DA_X} + \vec{X}^R_{AB_X}}{r_{DB}} \quad (196)$$

$$\frac{\delta r_{DB}}{\delta \vec{X}^R_{AB_Y}} = \frac{\vec{X}^R_{DA_Y} + \vec{X}^R_{AB_Y}}{r_{DB}} \quad (197)$$

$$\frac{\delta r_{DB}}{\delta \vec{X}^R_{AB_Z}} = \frac{\vec{X}^R_{DA_Z} + \vec{X}^R_{AB_Z}}{r_{DB}} \quad (198)$$

Partial derivative of r_{DB} with respect to $\vec{X}^R_{AC}$:

$$\frac{\delta r_{DB}}{\delta \vec{X}^R_{AC_X}} = \frac{\delta r_{DB}}{\delta \vec{X}^R_{AC_Y}} = \frac{\delta r_{DB}}{\delta \vec{X}^R_{AC_Z}} = \mathbf{0} \quad (199)$$

To calculate the value of θ_{DB}^{R}:

$$\boldsymbol{\theta_{DB}^{R} = -\sin^{-1}\left(\frac{\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}}{r_{DB}}\right)} \quad (200)$$

Partial derivative of θ_{DB}^{R} with respect to $\vec{X}_{RA}^{R}$:

$$\frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{RA_X}^{R}} = \frac{\left(\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}\right)*\left(\vec{X}_{DA_X}^{R} + \vec{X}_{AB_X}^{R}\right)}{r_{DB}^{3}\sqrt{1-\left(\frac{\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}}{r_{DB}}\right)^{2}}} \quad (201)$$

$$\frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{RA_Y}^{R}} = \frac{\left(\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}\right)*\left(\vec{X}_{DA_Y}^{R} + \vec{X}_{AB_Y}^{R}\right)}{r_{DB}^{3}\sqrt{1-\left(\frac{\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}}{r_{DB}}\right)^{2}}} \quad (202)$$

$$\frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{RA_Z}^{R}} = \frac{-r_{DB}^{2} + \left(\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}\right)^{2}}{r_{DB}^{3}\sqrt{1-\left(\frac{\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}}{r_{DB}}\right)^{2}}} \quad (203)$$

Partial derivative of θ_{DB}^{R} with respect to $\vec{V}_{RA}^{R}$:

$$\frac{\delta\theta_{DB}^{R}}{\delta\vec{V}_{RA_X}^{R}} = \frac{\delta\theta_{DB}^{R}}{\delta\vec{V}_{RA_Y}^{R}} = \frac{\delta\theta_{DB}^{R}}{\delta\vec{V}_{RA_Z}^{R}} = \mathbf{0} \quad (204)$$

Partial derivative of θ_{DB}^{R} with respect to $\vec{X}_{AB}^{R}$:

$$\frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{AB_X}^{R}} = \frac{\left(\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}\right)*\left(\vec{X}_{DA_X}^{R} + \vec{X}_{AB_X}^{R}\right)}{r_{DB}^{3}\sqrt{1-\left(\frac{\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}}{r_{DB}}\right)^{2}}} \quad (205)$$

$$\frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{AB_Y}^{R}} = \frac{\left(\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}\right)*\left(\vec{X}_{DA_Y}^{R} + \vec{X}_{AB_Y}^{R}\right)}{r_{DB}^{3}\sqrt{1-\left(\frac{\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}}{r_{DB}}\right)^{2}}} \quad (206)$$

$$\frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{AB_Z}^{R}} = \frac{-r_{DB}^{2}+\left(\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}\right)^{2}}{r_{DB}^{3}\sqrt{1-\left(\frac{\vec{X}_{DA_Z}^{R} + \vec{X}_{AB_Z}^{R}}{r_{DB}}\right)^{2}}} \quad (207)$$

Partial derivative of θ_{DB}^{R} with respect to $\vec{X}_{AC}^{R}$:

$$\frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{AC_X}^{R}} = \frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{AC_Y}^{R}} = \frac{\delta\theta_{DB}^{R}}{\delta\vec{X}_{AC_Z}^{R}} = \mathbf{0} \quad (208)$$

Find ψ_{DC}^{R} and θ_{DC}^{R} mapping equations:

$$\boldsymbol{H_{31}} = \begin{bmatrix} \frac{\delta\psi_{DC}^{R}}{\delta\vec{X}_{RA_X}^{R}} & \frac{\delta\psi_{DC}^{R}}{\delta\vec{X}_{RA_Y}^{R}} & \frac{\delta\psi_{DC}^{R}}{\delta\vec{X}_{RA_z}^{R}} \\ \frac{\delta\theta_{DC}^{R}}{\delta\vec{X}_{RA_X}^{R}} & \frac{\delta\theta_{DC}^{R}}{\delta\vec{X}_{RA_Y}^{R}} & \frac{\delta\theta_{DC}^{R}}{\delta\vec{X}_{RA_z}^{R}} \end{bmatrix} \quad (209)$$

$$\boldsymbol{H_{32}} = \begin{bmatrix} \frac{\delta \psi_{DC}^{R}}{\delta \vec{V}_{RA_X}^{R}} & \frac{\delta \psi_{DC}^{R}}{\delta \vec{V}_{RA_Y}^{R}} & \frac{\delta \psi_{DC}^{R}}{\delta \vec{V}_{RA_Z}^{R}} \\ \frac{\delta \theta_{DC}^{R}}{\delta \vec{V}_{RA_X}^{R}} & \frac{\delta \theta_{DC}^{R}}{\delta \vec{V}_{RA_Y}^{R}} & \frac{\delta \theta_{DC}^{R}}{\delta \vec{V}_{RA_Z}^{R}} \end{bmatrix} \quad (210)$$

$$\boldsymbol{H_{33}} = \begin{bmatrix} \frac{\delta \psi_{DC}^{R}}{\delta \vec{X}_{AB_X}^{R}} & \frac{\delta \psi_{DC}^{R}}{\delta \vec{X}_{AB_Y}^{R}} & \frac{\delta \psi_{DC}^{R}}{\delta \vec{X}_{AB_Z}^{R}} \\ \frac{\delta \theta_{DC}^{R}}{\delta \vec{X}_{AB_X}^{R}} & \frac{\delta \theta_{DC}^{R}}{\delta \vec{X}_{AB_Y}^{R}} & \frac{\delta \theta_{DC}^{R}}{\delta \vec{X}_{AB_Z}^{R}} \end{bmatrix} \quad (211)$$

$$\boldsymbol{H_{34}} = \begin{bmatrix} \frac{\delta \psi_{DC}^{R}}{\delta \vec{X}_{AC_X}^{R}} & \frac{\delta \psi_{DC}^{R}}{\delta \vec{X}_{AC_Y}^{R}} & \frac{\delta \psi_{DC}^{R}}{\delta \vec{X}_{AC_Z}^{R}} \\ \frac{\delta \theta_{DC}^{R}}{\delta \vec{X}_{AC_X}^{R}} & \frac{\delta \theta_{DC}^{R}}{\delta \vec{X}_{AC_Y}^{R}} & \frac{\delta \theta_{DC}^{R}}{\delta \vec{X}_{AC_Z}^{R}} \end{bmatrix} \quad (212)$$

The equation for ψ_{DC}^{R}:

$$\boldsymbol{\psi_{DC}^{R}} = \mathbf{tan^{-1}}\left(\frac{\vec{X}_{DA_Y}^{R} + \vec{X}_{AC_Y}^{R}}{\vec{X}_{DA_X}^{R} + \vec{X}_{AC_X}^{R}}\right) \quad (213)$$

Partial derivative of ψ_{DC}^{R} with respect to $\vec{X}_{RA}^{R}$:

$$\frac{\delta \psi_{DC}^{R}}{\delta \vec{X}_{RA_X}^{R}} = \frac{-\left(\vec{X}_{DA_Y}^{R} + \vec{X}_{AC_Y}^{R}\right)}{\left(\vec{X}_{DA_X}^{R} + \vec{X}_{AC_X}^{R}\right)^2 + \left(\vec{X}_{DA_Y}^{R} + \vec{X}_{AC_Y}^{R}\right)^2} \quad (214)$$

$$\frac{\delta \psi_{DC}^{R}}{\delta \vec{X}_{RA_Y}^{R}} = \frac{\vec{X}_{DA_X}^{R} + \vec{X}_{AC_X}^{R}}{\left(\vec{X}_{DA_X}^{R} + \vec{X}_{AC_X}^{R}\right)^2 + \left(\vec{X}_{DA_Y}^{R} + \vec{X}_{AC_Y}^{R}\right)^2} \quad (215)$$

$$\frac{\delta \psi_{DC}^{R}}{\delta \vec{X}_{RA_z}^{R}} = \mathbf{0} \tag{216}$$

Partial derivative of ψ_{DC}^{R} with respect to $\vec{V}_{RA}^{R}$:

$$\frac{\delta \psi_{DC}^{R}}{\delta \vec{V}_{RA_X}^{R}} = \frac{\delta \psi_{DC}^{R}}{\delta \vec{V}_{RA_Y}^{R}} = \frac{\delta \psi_{DC}^{R}}{\delta \vec{V}_{RA_Z}^{R}} = \mathbf{0} \tag{217}$$

Partial derivative of ψ_{DC}^{R} with respect to $\vec{X}_{AB}^{R}$:

$$\frac{\delta \psi_{DC}^{R}}{\delta \vec{X}_{AB_X}^{R}} = \frac{\delta \psi_{DC}^{R}}{\delta \vec{X}_{AB_Y}^{R}} = \frac{\delta \psi_{DC}^{R}}{\delta \vec{X}_{AB_Z}^{R}} = \mathbf{0} \tag{218}$$

Partial derivative of ψ_{DC}^{R} with respect to $\vec{X}_{AC}^{R}$:

$$\frac{\delta \psi_{DC}^{R}}{\delta \vec{X}_{AC_X}^{R}} = \frac{-\left(\vec{X}_{DA_Y}^{R} + \vec{X}_{AC_Y}^{R}\right)}{\left(\vec{X}_{DA_X}^{R} + \vec{X}_{AC_X}^{R}\right)^2 + \left(\vec{X}_{DA_Y}^{R} + \vec{X}_{AC_Y}^{R}\right)^2} \tag{219}$$

$$\frac{\delta \psi_{DC}^{R}}{\delta \vec{X}_{AC_Y}^{R}} = \frac{\vec{X}_{DA_X}^{R} + \vec{X}_{AB_X}^{R}}{\left(\vec{X}_{DA_X}^{R} + \vec{X}_{AC_X}^{R}\right)^2 + \left(\vec{X}_{DA_Y}^{R} + \vec{X}_{AC_Y}^{R}\right)^2} \tag{220}$$

$$\frac{\delta \psi_{DC}^{R}}{\delta \vec{X}_{AC_z}^{R}} = \mathbf{0} \tag{221}$$

Equation to calculater_{DC}:

$$\boldsymbol{r_{DC}} = \sqrt{\left(\vec{\boldsymbol{X}}^{\boldsymbol{R}}_{\boldsymbol{DA_X}} + \vec{\boldsymbol{X}}^{\boldsymbol{R}}_{\boldsymbol{AC_X}}\right)^2 + \left(\vec{\boldsymbol{X}}^{\boldsymbol{R}}_{\boldsymbol{DA_Y}} + \vec{\boldsymbol{X}}^{\boldsymbol{R}}_{\boldsymbol{AC_Y}}\right)^2 + \left(\vec{\boldsymbol{X}}^{\boldsymbol{R}}_{\boldsymbol{DA_Z}} + \vec{\boldsymbol{X}}^{\boldsymbol{R}}_{\boldsymbol{AC_Z}}\right)^2} \quad (222)$$

Partial derivative of r_{DC} with respect to $\vec{X}^R_{RA}$:

$$\frac{\boldsymbol{\delta r_{DC}}}{\boldsymbol{\delta \vec{X}^R_{RA_X}}} = \frac{\vec{\boldsymbol{X}}^{\boldsymbol{R}}_{\boldsymbol{DA_X}} + \vec{\boldsymbol{X}}^{\boldsymbol{R}}_{\boldsymbol{AC_X}}}{\boldsymbol{r_{DC}}} \quad (223)$$

$$\frac{\boldsymbol{\delta r_{DC}}}{\boldsymbol{\delta \vec{X}^R_{RA_Y}}} = \frac{\vec{\boldsymbol{X}}^{\boldsymbol{R}}_{\boldsymbol{DA_Y}} + \vec{\boldsymbol{X}}^{\boldsymbol{R}}_{\boldsymbol{AC_Y}}}{\boldsymbol{r_{DC}}} \quad (224)$$

$$\frac{\boldsymbol{\delta r_{DC}}}{\boldsymbol{\delta \vec{X}^R_{RA_Z}}} = \frac{\vec{\boldsymbol{X}}^{\boldsymbol{R}}_{\boldsymbol{DA_Z}} + \vec{\boldsymbol{X}}^{\boldsymbol{R}}_{\boldsymbol{AC_Z}}}{\boldsymbol{r_{DC}}} \quad (225)$$

Partial derivative of r_{DC} with respect to $\vec{V}^R_{RA}$:

$$\frac{\boldsymbol{\delta r_{DC}}}{\boldsymbol{\delta \vec{V}^R_{RA_X}}} = \frac{\boldsymbol{\delta r_{DC}}}{\boldsymbol{\delta \vec{V}^R_{RA_Y}}} = \frac{\boldsymbol{\delta r_{DC}}}{\boldsymbol{\delta \vec{V}^R_{RA_Z}}} = \boldsymbol{0} \quad (226)$$

Partial derivative of r_{DC} with respect to $\vec{X}^R_{AB}$:

$$\frac{\boldsymbol{\delta r_{DC}}}{\boldsymbol{\delta \vec{X}^R_{AB_X}}} = \frac{\boldsymbol{\delta r_{DC}}}{\boldsymbol{\delta \vec{X}^R_{AB_Y}}} = \frac{\boldsymbol{\delta r_{DC}}}{\boldsymbol{\delta \vec{X}^R_{AB_Z}}} = \boldsymbol{0} \quad (227)$$

Partial derivative of r_{DC} with respect to $\vec{X}_{AC}^{R}$:

$$\frac{\delta r_{DC}}{\delta \vec{X}_{AC_X}^{R}} = \frac{\vec{X}_{DA_X}^{R} + \vec{X}_{AC_X}^{R}}{r_{DC}} \quad (228)$$

$$\frac{\delta r_{DC}}{\delta \vec{X}_{AC_Y}^{R}} = \frac{\vec{X}_{DA_Y}^{R} + \vec{X}_{AC_Y}^{R}}{r_{DC}} \quad (229)$$

$$\frac{\delta r_{DC}}{\delta \vec{X}_{AC_Z}^{R}} = \frac{\vec{X}_{DA_Z}^{R} + \vec{X}_{AC_Z}^{R}}{r_{DC}} \quad (230)$$

Equation to calculate θ_{DC}^{R}:

$$\boldsymbol{\theta_{DC}^{R}} = -\sin^{-1}\left(\frac{\vec{X}_{DA_Z}^{R} + \vec{X}_{AC_Z}^{R}}{r_{DC}}\right) \quad (231)$$

Partial derivative of θ_{DC}^{R} with respect to $\vec{X}_{RA}^{R}$:

$$\frac{\delta \theta_{DC}^{R}}{\delta \vec{X}_{RA_X}^{R}} = \frac{\left(\vec{X}_{DA_Z}^{R} + \vec{X}_{AC_Z}^{R}\right) * \left(\vec{X}_{DA_X}^{R} + \vec{X}_{AC_X}^{R}\right)}{r_{DC}^{3}\sqrt{1 - \left(\frac{\vec{X}_{DA_Z}^{R} + \vec{X}_{AC_Z}^{R}}{r_{DC}}\right)^{2}}} \quad (232)$$

$$\frac{\delta \theta_{DC}^{R}}{\delta \vec{X}_{RA_Y}^{R}} = \frac{\left(\vec{X}_{DA_Z}^{R} + \vec{X}_{AC_Z}^{R}\right) * \left(\vec{X}_{DA_Y}^{R} + \vec{X}_{AC_Y}^{R}\right)}{r_{DC}^{3}\sqrt{1 - \left(\frac{\vec{X}_{DA_Z}^{R} + \vec{X}_{AC_Z}^{R}}{r_{DC}}\right)^{2}}} \quad (233)$$

$$\frac{\delta\theta_{DC}^{R}}{\delta\vec{X}_{RA_Z}^{R}} = \frac{-r_{DC}^{2}+\left(\vec{X}_{DA_Z}^{R}+\vec{X}_{AC_Z}^{R}\right)^{2}}{r_{DC}^{3}\sqrt{1-\left(\frac{\vec{X}_{DA_Z}^{R}+\vec{X}_{AC_Z}^{R}}{r_{DC}}\right)^{2}}} \quad (234)$$

Partial derivative of θ_{DC}^{R} with respect to $\vec{V}_{RA}^{R}$:

$$\frac{\delta\theta_{DC}^{R}}{\delta\vec{V}_{RA_X}^{R}} = \frac{\delta\theta_{DC}^{R}}{\delta\vec{V}_{RA_Y}^{R}} = \frac{\delta\theta_{DC}^{R}}{\delta\vec{V}_{RA_Z}^{R}} = \mathbf{0} \quad (235)$$

Partial derivative of θ_{DC}^{R} with respect to $\vec{X}_{AB}^{R}$:

$$\frac{\delta\theta_{DC}^{R}}{\delta\vec{X}_{AB_X}^{R}} = \frac{\delta\theta_{DC}^{R}}{\delta\vec{X}_{AB_Y}^{R}} = \frac{\delta\theta_{DC}^{R}}{\delta\vec{X}_{AB_Z}^{R}} = \mathbf{0} \quad (236)$$

Partial derivative of θ_{DC}^{R} with respect to $\vec{X}_{AC}^{R}$:

$$\frac{\delta\theta_{DC}^{R}}{\delta\vec{X}_{AC_X}^{R}} = \frac{\left(\vec{X}_{DA_Z}^{R}+\vec{X}_{AC_Z}^{R}\right)*\left(\vec{X}_{DA_X}^{R}+\vec{X}_{AC_X}^{R}\right)}{r_{DC}^{3}\sqrt{1-\left(\frac{\vec{X}_{DA_Z}^{R}+\vec{X}_{AC_Z}^{R}}{r_{DC}}\right)^{2}}} \quad (237)$$

$$\frac{\delta\theta_{DC}^{R}}{\delta\vec{X}_{AC_Y}^{R}} = \frac{\left(\vec{X}_{DA_Z}^{R}+\vec{X}_{AC_Z}^{R}\right)*\left(\vec{X}_{DA_Y}^{R}+\vec{X}_{AC_Y}^{R}\right)}{r_{DC}^{3}\sqrt{1-\left(\frac{\vec{X}_{DA_Z}^{R}+\vec{X}_{AC_Z}^{R}}{r_{DC}}\right)^{2}}} \quad (238)$$

$$\frac{\delta\theta_{DC}^{R}}{\delta\vec{X}_{AC_Z}^{R}} = \frac{-r_{DC}^{2}+\left(\vec{X}_{DA_Z}^{R}+\vec{X}_{AC_Z}^{R}\right)^{2}}{r_{DC}^{3}\sqrt{1-\left(\frac{\vec{X}_{DA_Z}^{R}+\vec{X}_{AC_Z}^{R}}{r_{DC}}\right)^{2}}} \quad (239)$$

Native Spatial to PCA Domain

The transformation of the vector angle and magnitude features from the Native Spatial domain to the PCA domain was achieved by projecting the Native Spatial data onto the principal eigenvectors calculated from the training data.

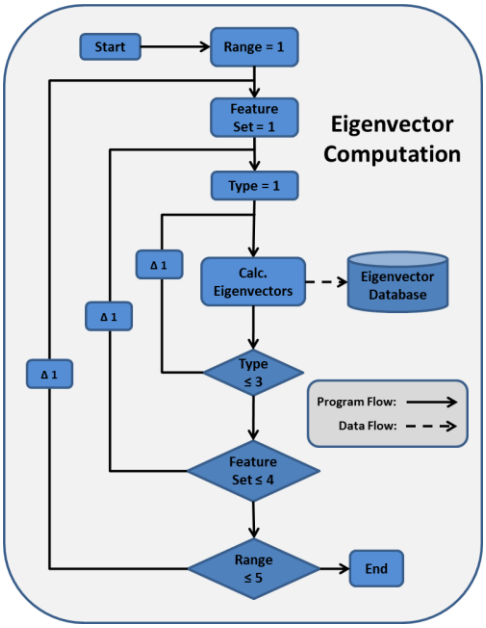

Figure 13 Eigenvector Generation

As shown in Figure 13, Eigenvectors are calculated for each range, feature set and vehicle type and applied to the unknown data before classification. The number of Principal Components used for each approach is listed in Table 3.

Table 3
Number Of Principal Components Used For Each Approach

Approach	# Of Principal Components
2 Feature	2
3 Feature	3
7 Feature	6
16 Feature	8

Native Spatial to LDA Domain

The transformation of the vector angle and magnitude features from the spatial domain to the LDA domain was achieved by multiplying the LDA weighting matrix with the spatial data matrix. The LDA weighting matrix was calculated as shown in Figure 14.

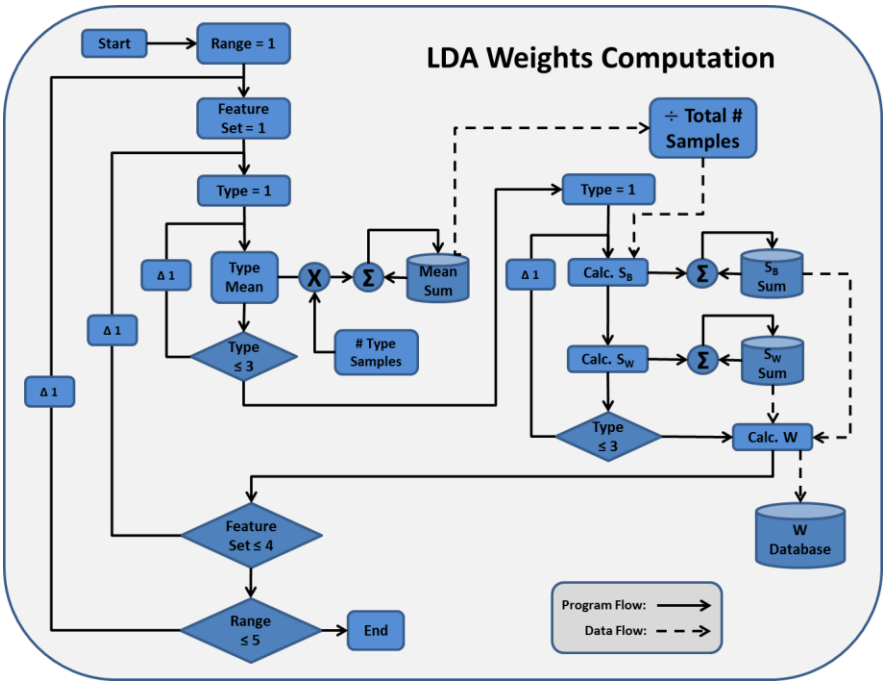

Figure 14 LDA Weight Computation

CHAPTER FOUR:
TWO FEATURE CLASSIFICATION OF VEHICLES

Using the dot product formula to calculate the relative angel between two vectors provides a convenient mechanism for characterizing the orientation between those vectors. For the proposed two feature classification technique, the vectors or interest are the vector between PoI *A* and PoI *B* on the vehicle and the velocity vector of PoI *A*. The velocity vector of PoI *A* is assumed to have a constant orientation to the vector between PoI *A* and PoI *B* on the vehicle regardless of vehicle orientation. It's worth re-noting, one of the requirements for the proposed techniques is the velocity vector of the vehicle is constant. In other words, the vehicle can't be turning or changing lanes. Given that the velocity vector is being used a reference vector, the two feature approach is only applicable to moving vehicles. Equations 240 and 241 demonstrate the calculation of the features used for the two feature classification approach. The features are derived either from the dot product angle or the length of the vectors.

$$\boldsymbol{r_{AB}} = \sqrt{\left(\vec{\boldsymbol{X}}_{\boldsymbol{B}}^{\boldsymbol{R}} - \vec{\boldsymbol{X}}_{\boldsymbol{A}}^{\boldsymbol{R}}\right)^{\boldsymbol{2}}} \qquad (240)$$

$$\boldsymbol{\gamma_{AB}} = \boldsymbol{cos^{-1}}\left(\frac{\vec{\boldsymbol{X}}_{\boldsymbol{AB}}^{\boldsymbol{R}} \cdot \vec{\boldsymbol{V}}_{\boldsymbol{A}}^{\boldsymbol{R}}}{|\vec{\boldsymbol{X}}_{\boldsymbol{AB}}^{\boldsymbol{R}}| * |\vec{\boldsymbol{V}}_{\boldsymbol{A}}^{\boldsymbol{R}}|}\right) \qquad (241)$$

The two feature classification approach was implemented using two separate Kalman Filter formulations. The first formulation, referred to as Filter Type 1, employed two separate six state filters which each independently tracked a point of interest. Using the 3D

position information contained in each six state filter for their respective point of interest, the vectors and angles of interest were calculated. The second formulation, referred to as Filter Type 2, employed a single nine state filter which uses knowledge of the feature geometry to track both PoI as a single rigid structure. Using the 3D position information for the PoI contained in the nine state filter, the vectors and angles of interest were calculated. The symbols corresponding to the different domain data types tested in each classifier are depicted in Figure 15.

Data Domain	Symbol
Native Spatial	
PCA	
LDA	
Ensemble	

Figure 15 Data Domain Two Feature Classification Plot Symbols

Experimental Results – Filter Type 1 - Six State Filters

The classification performance of Filter Type 1 for the two feature approach is detailed in this section for constant velocity vehicles with varying levels of noise.

Noise Level of 50 microradians

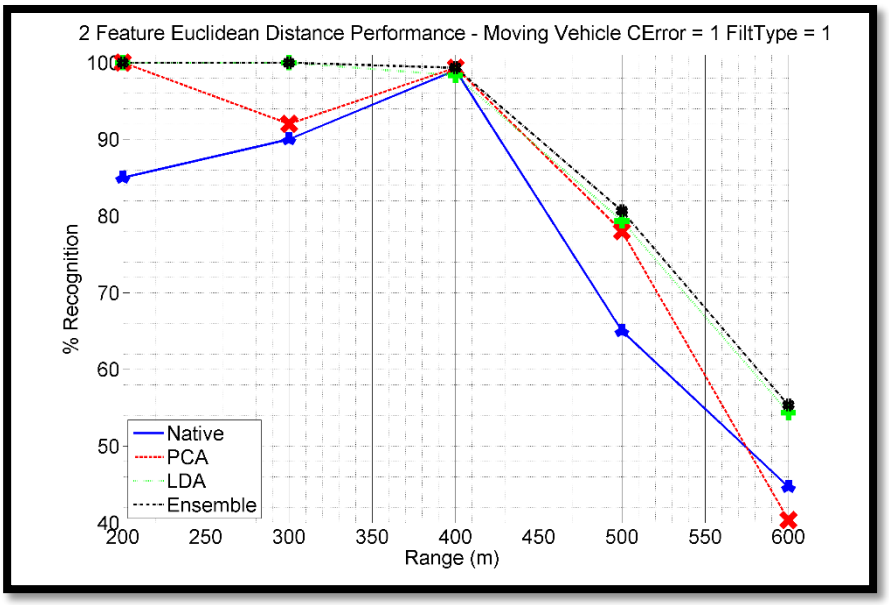

Figure 16 Percent Recognition of Vehicle Types -vs- Range with Two Feature Euclidean Distance: Moving Vehicle - Noise Level of 50 microradians - Filter Type 1

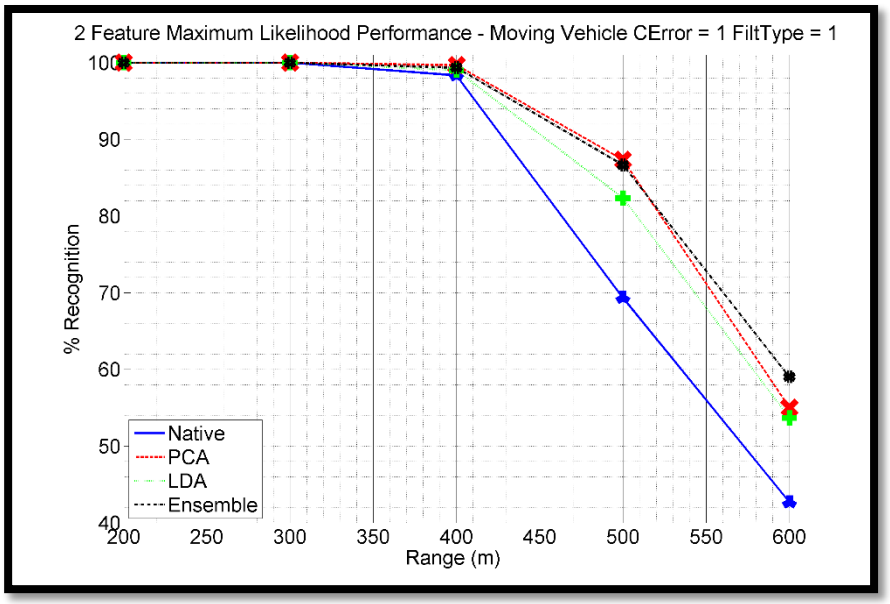

Figure 17 Percent Recognition of Vehicle Types -vs- Range with Two Feature Max Likelihood: Moving Vehicle - Noise Level of 50 microradians- Filter Type 1

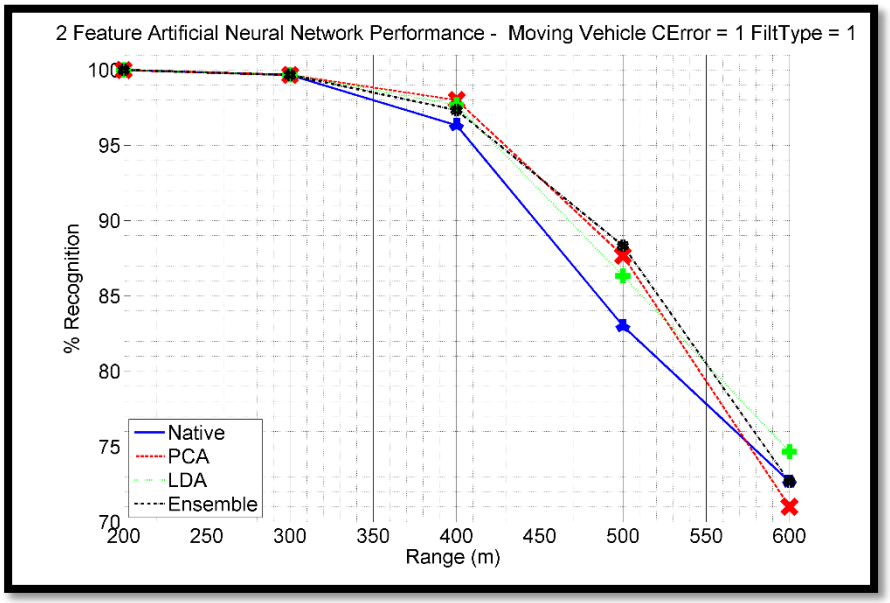

Figure 18 Percent Recognition of Vehicle Types -vs- Range with Two Feature ANN: Moving Vehicle - Noise Level of 50 microradians - Filter Type 1

Noise Level of 100 microradians

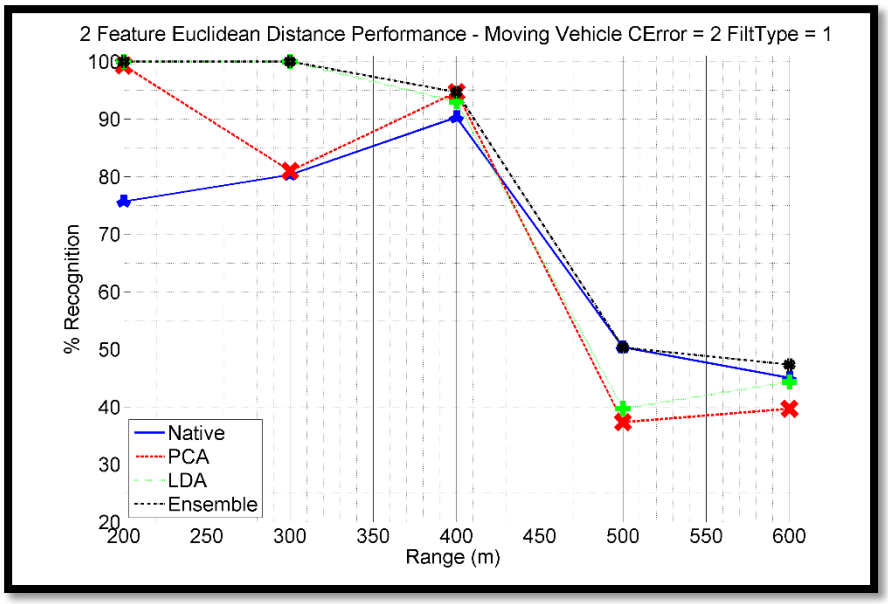

Figure 19 Percent Recognition of Vehicle Types -vs- Range with Two Feature Euclidean Distance: Moving Vehicle - Noise Level of 100 microradians - Filter Type 1

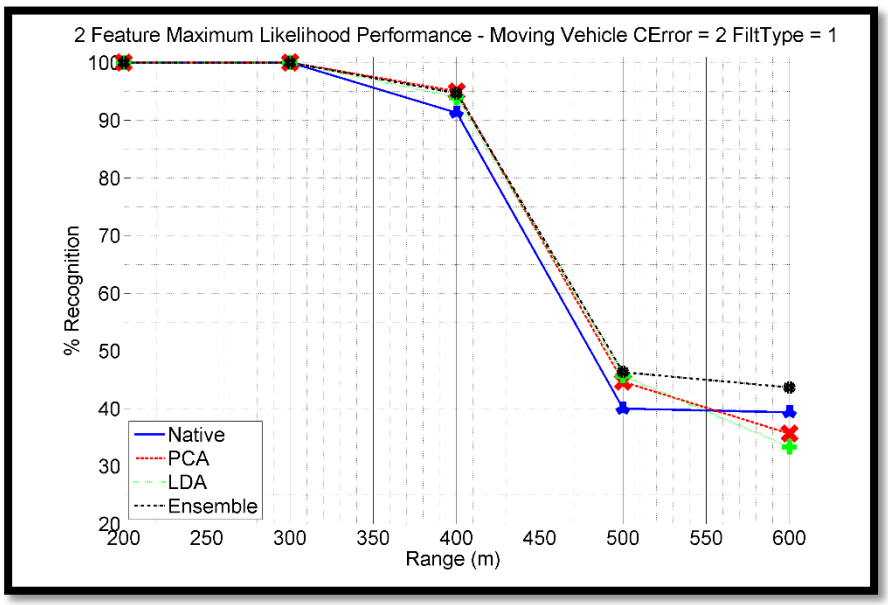

Figure 20 Percent Recognition of Vehicle Types -vs- Range with Two Feature Max Likelihood: Moving Vehicle - Noise Level of 100 microradians – Filter Type 1

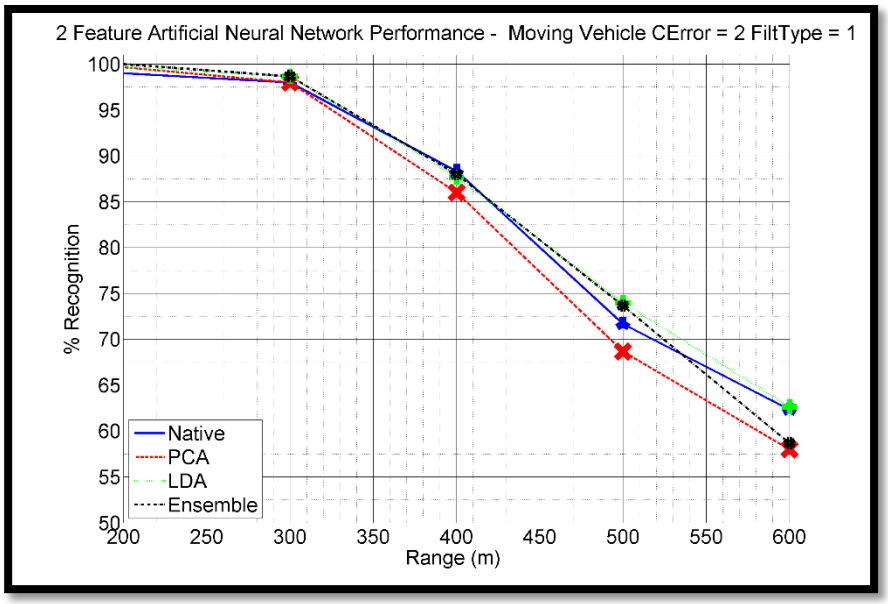

Figure 21 Percent Recognition of Vehicle Types -vs- Range with Two Feature ANN: Moving Vehicle - Noise Level of 100 microradians - Filter Type 1

Noise Level of 150 microradians

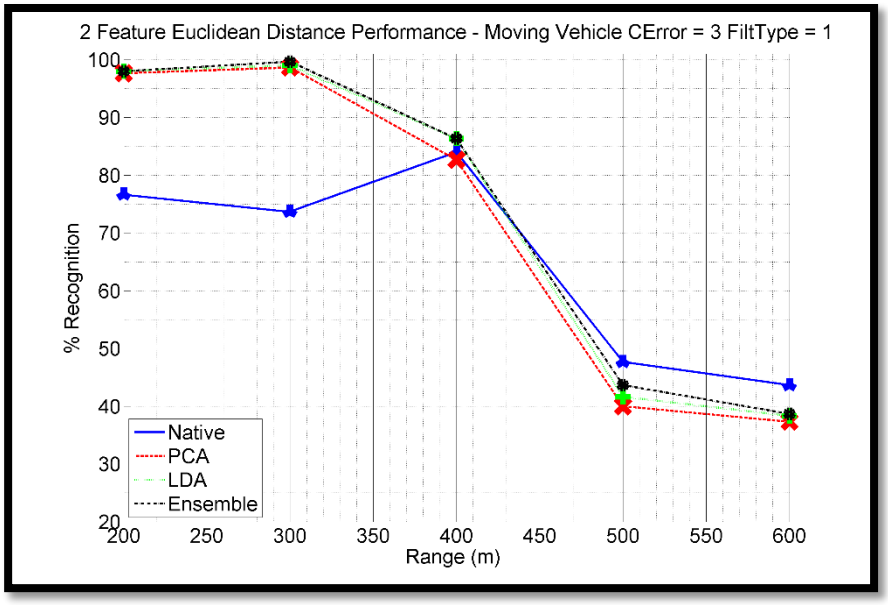

Figure 22 Percent Recognition of Vehicle Types -vs- Range with Two Feature Euclidean Distance: Moving Vehicle - Noise Level of 150 microradians - Filter Type 1

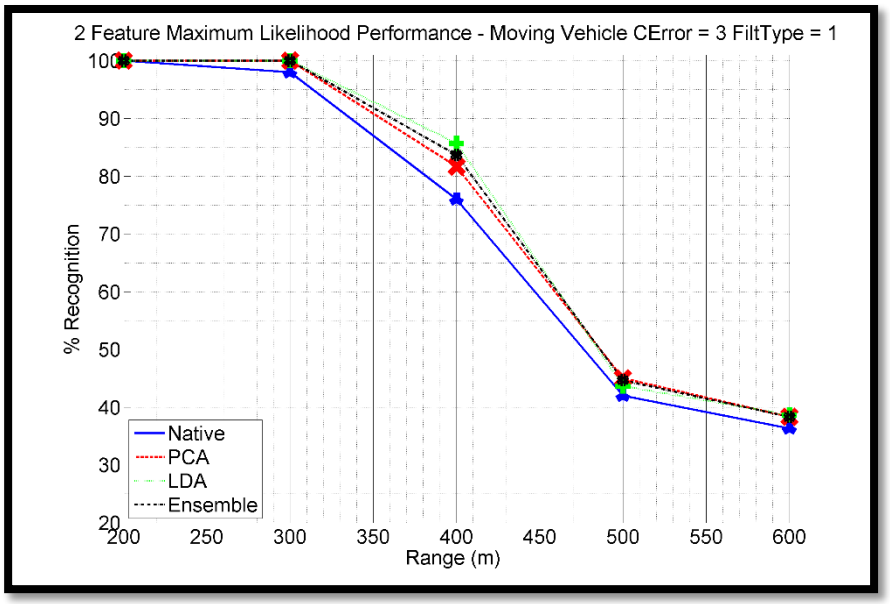

Figure 23 Percent Recognition of Vehicle Types -vs- Range with Two Feature Max Likelihood: Moving Vehicle - Noise Level of 150 microradians – Filter Type 1

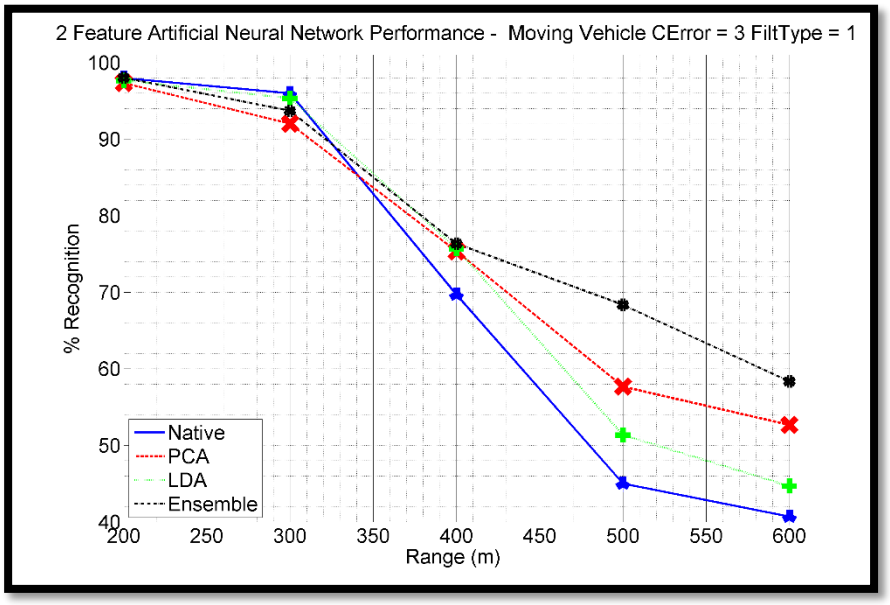

Figure 24 Percent Recognition of Vehicle Types -vs- Range with Two Feature ANN: Moving Vehicle - Noise Level of 150 microradians - Filter Type 1

Noise Level of 200 microradians

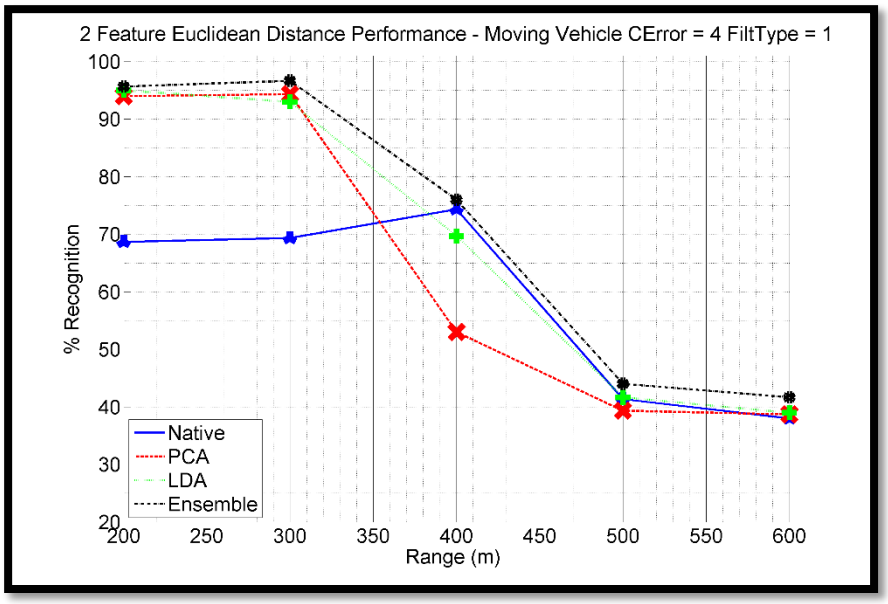

Figure 25 Percent Recognition of Vehicle Types -vs- Range with Two Feature Euclidean Distance: Moving Vehicle - Noise Level of 200 microradians - Filter Type 1

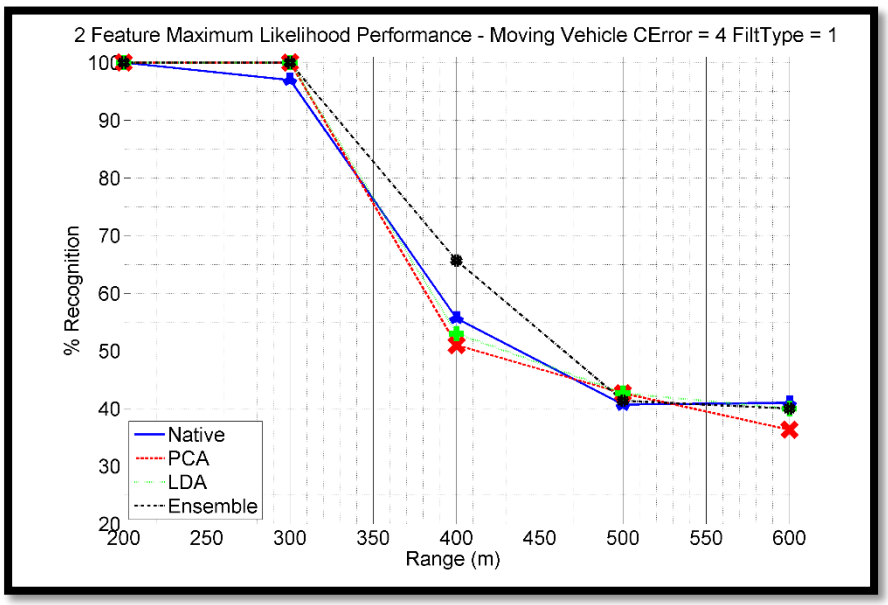

Figure 26 Percent Recognition of Vehicle Types -vs- Range with Two Feature Max Likelihood: Moving Vehicle - Noise Level of 200 microradians – Filter Type 1

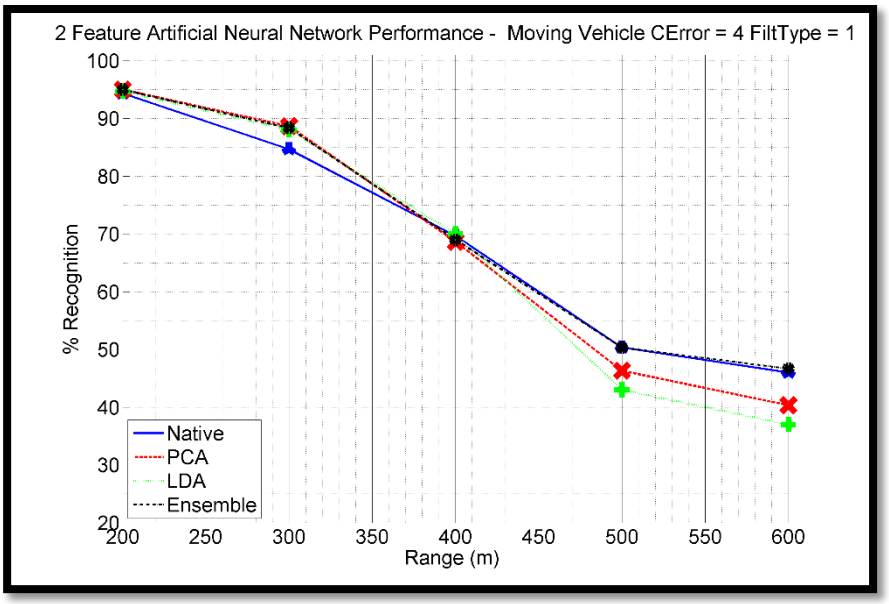

Figure 27 Percent Recognition of Vehicle Types -vs- Range with Two Feature ANN: Moving Vehicle - Noise Level of 200 microradians - Filter Type 1

Experimental Results – Filter Type 2 - Nine State Filter

The classification performance of Filter Type 2 for the two feature appraoch is detailed in this section for constant velocity vehicles with varying levels of noise.

Noise Level of 50 microradians

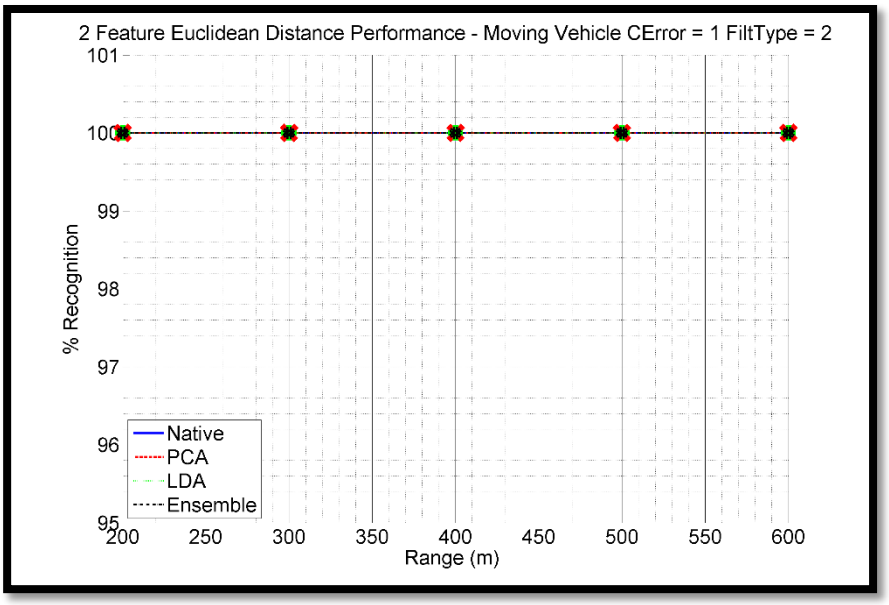

Figure 28 Percent Recognition of Vehicle Types -vs- Range with Two Feature Euclidean Distance: Moving Vehicle - Noise Level of 50 microradians - Filter Type 2

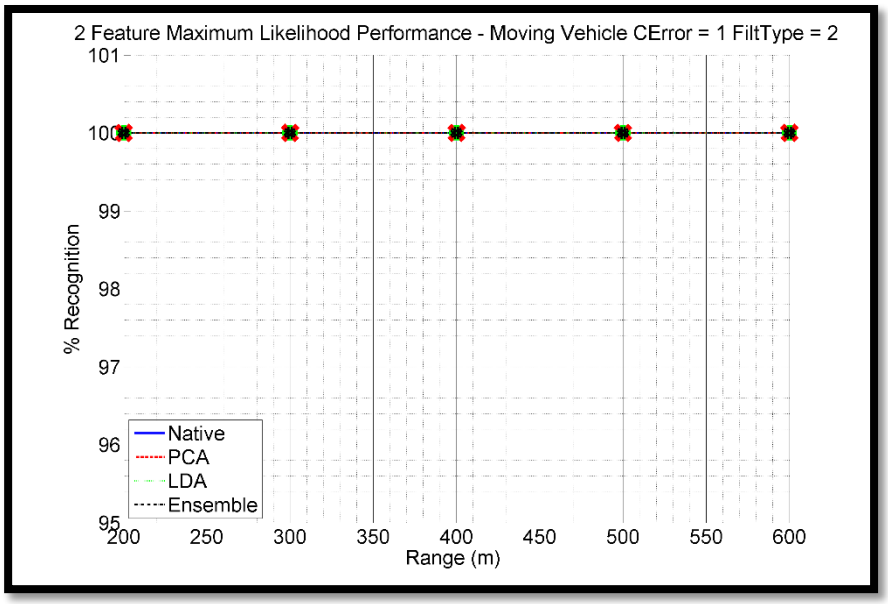

Figure 29 Percent Recognition of Vehicle Types -vs- Range with Two Feature Max Likelihood: Moving Vehicle - Noise Level of 50 microradians – Filter Type 2

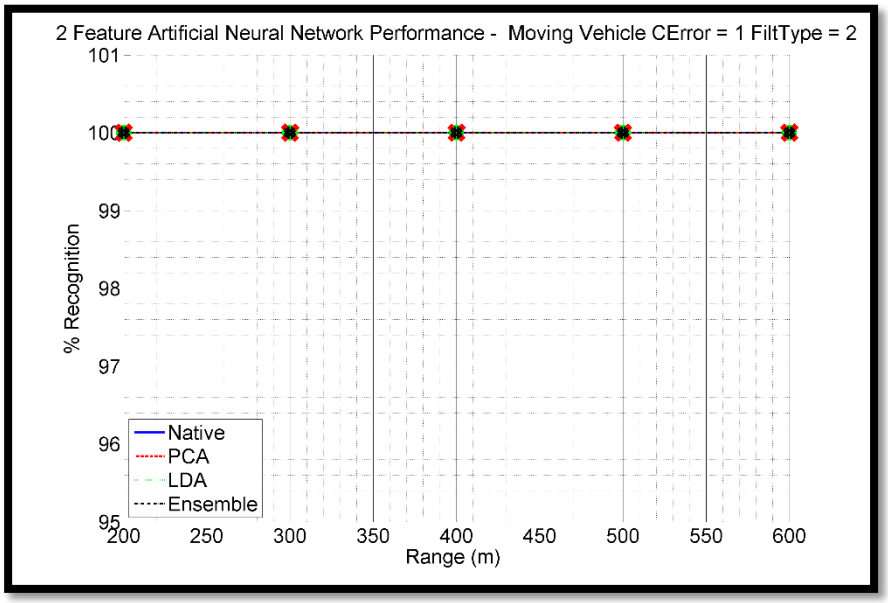

Figure 30 Percent Recognition of Vehicle Types -vs- Range with Two Feature ANN: Moving Vehicle - Noise Level of 50 microradians - Filter Type 2

Noise Level of 100 microradians

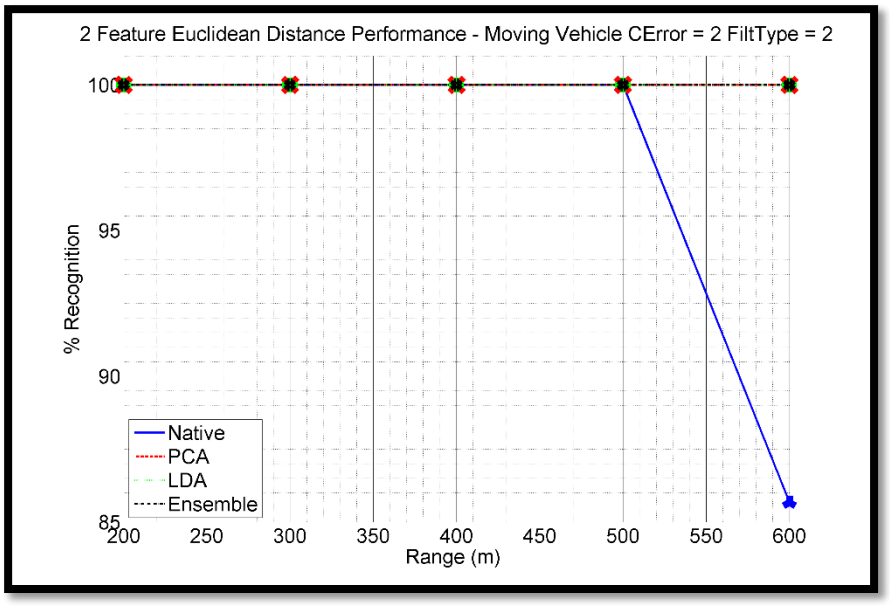

Figure 31 Percent Recognition of Vehicle Types -vs- Range with Two Feature Euclidean Distance: Moving Vehicle - Noise Level of 100 microradians - Filter Type 2

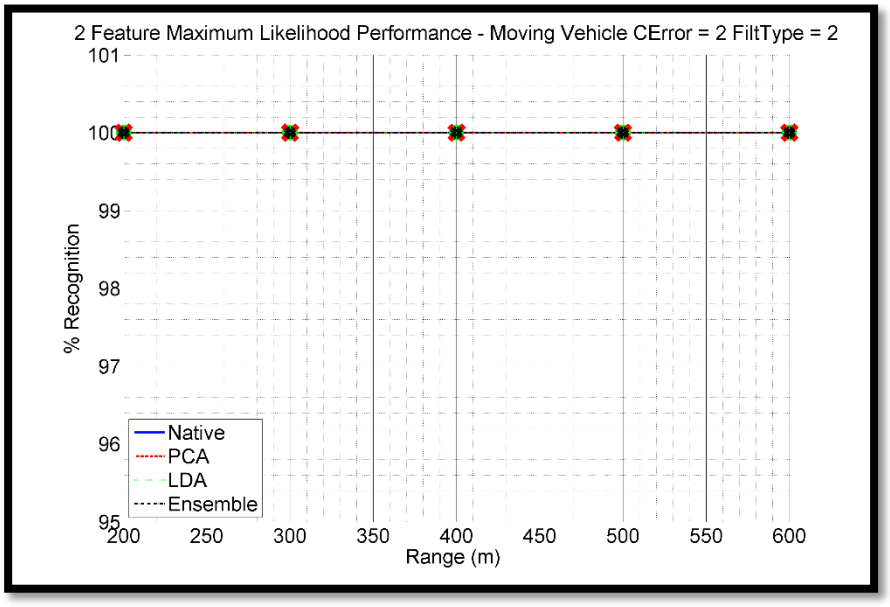

Figure 32 Percent Recognition of Vehicle Types -vs- Range with Two Feature Max Likelihood: Moving Vehicle - Noise Level of 100 microradians – Filter Type 2

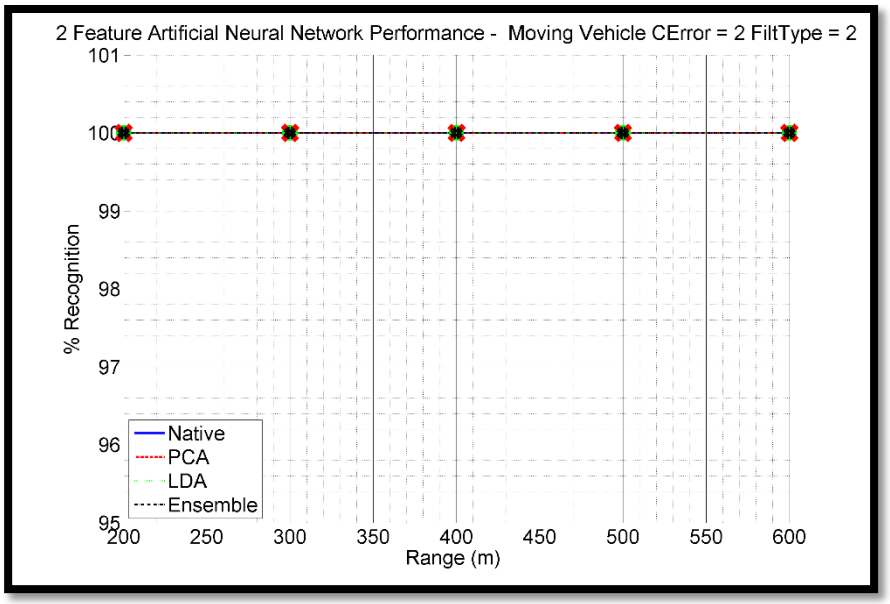

Figure 33 Percent Recognition of Vehicle Types -vs- Range with Two Feature ANN: Moving Vehicle - Noise Level of 100 microradians - Filter Type 2

Noise Level of 150 microradians

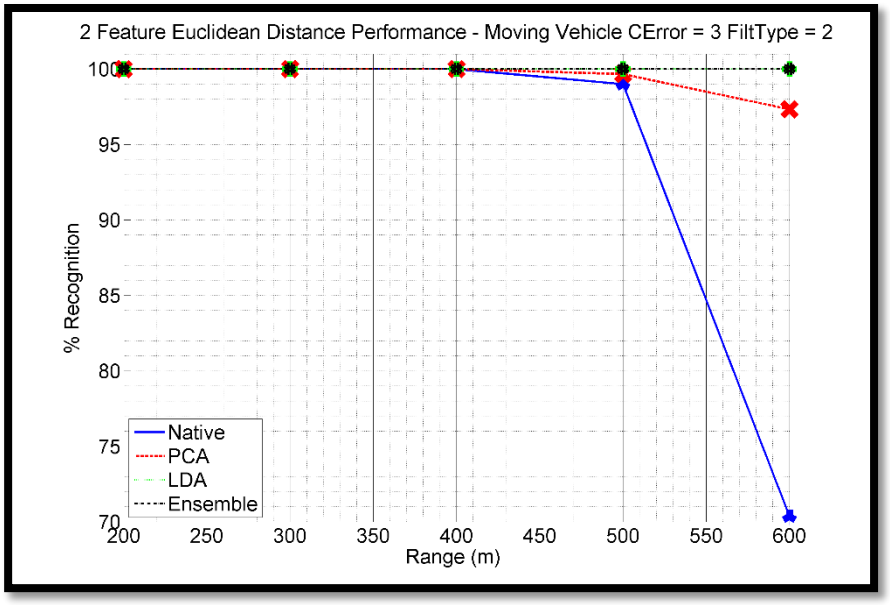

Figure 34 Percent Recognition of Vehicle Types -vs- Range with Two Feature Euclidean Distance: Moving Vehicle - Noise Level of 150 microradians - Filter Type 2

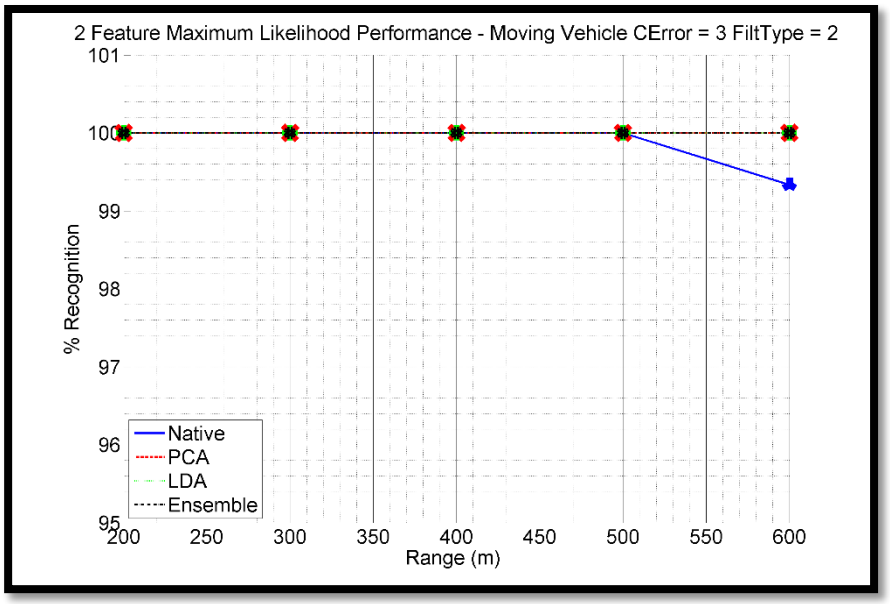

Figure 35 Percent Recognition of Vehicle Types -vs- Range with Two Feature Max Likelihood: Moving Vehicle - Noise Level of 150 microradians – Filter Type 2

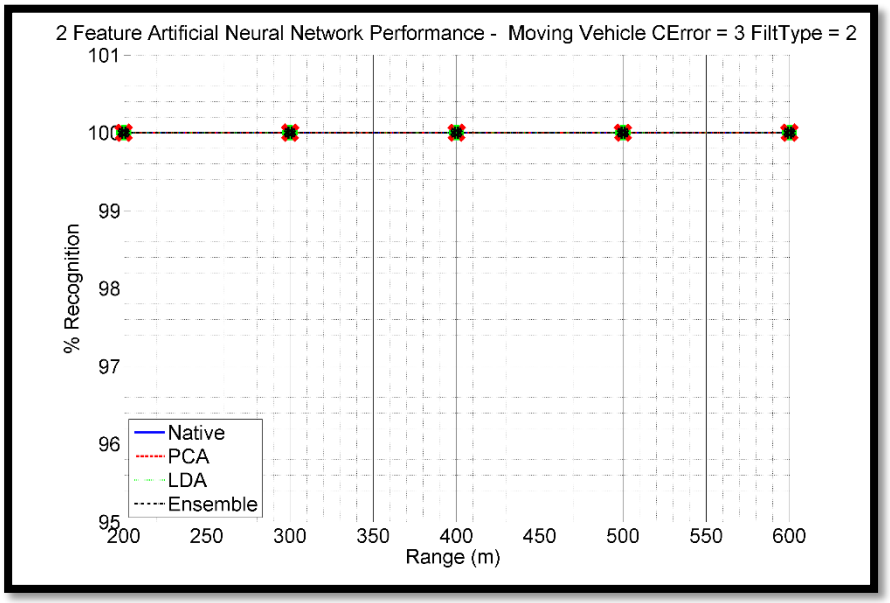

Figure 36 Percent Recognition of Vehicle Types -vs- Range with Two Feature ANN: Moving Vehicle - Noise Level of 150 microradians - Filter Type 2

Noise Level of 200 microradians

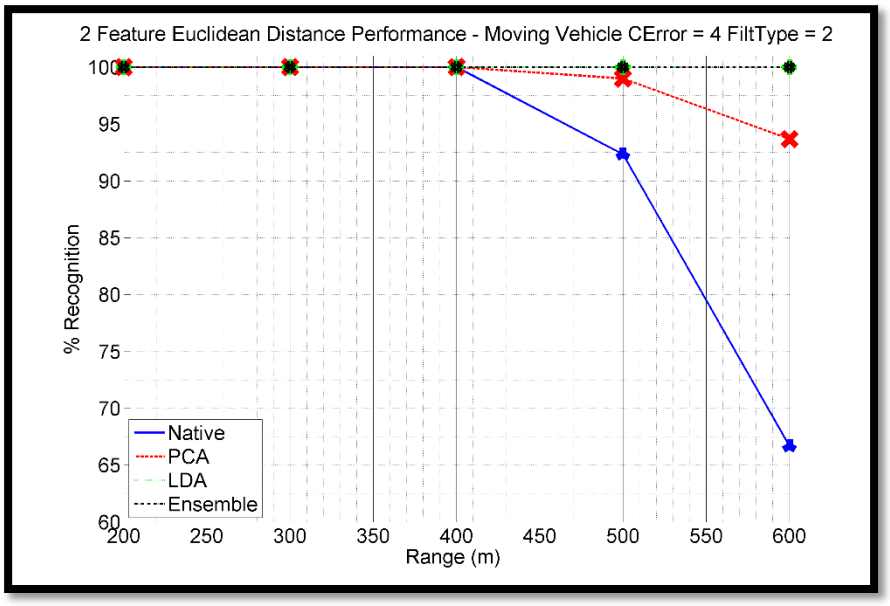

Figure 37 Percent Recognition of Vehicle Types -vs- Range with Two Feature Euclidean Distance: Moving Vehicle - Noise Level of 200 microradians - Filter Type 2

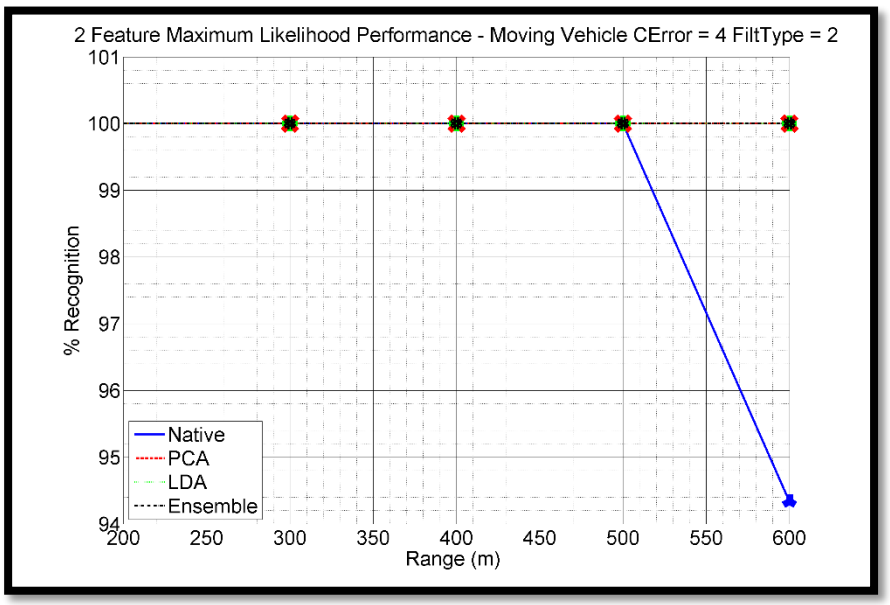

Figure 38 Percent Recognition of Vehicle Types -vs- Range with Two Feature Max Likelihood: Moving Vehicle - Noise Level of 200 microradians – Filter Type 2

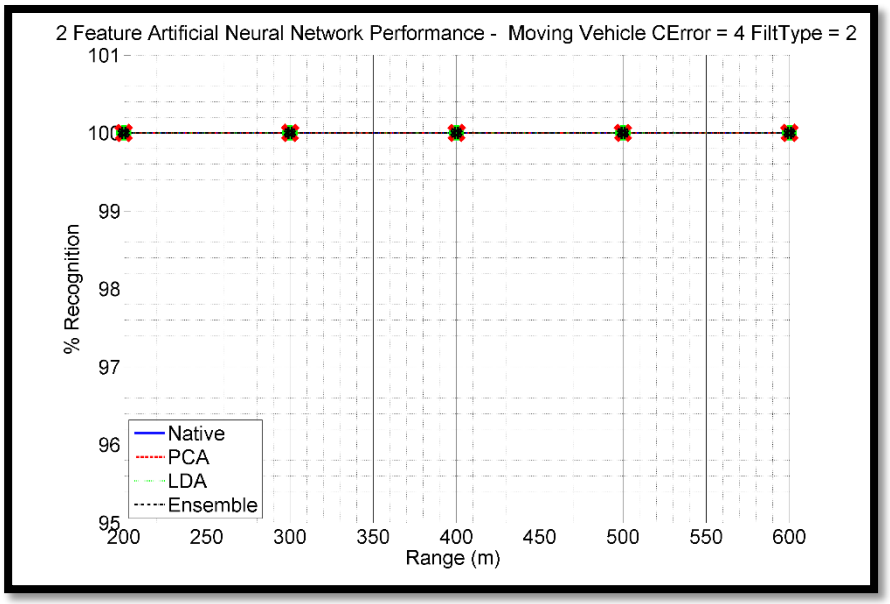

Figure 39 Percent Recognition of Vehicle Types -vs- Range with Two Feature ANN: Moving Vehicle - Noise Level of 200 microradians - Filter Type 2

Two Feature Conclusions

The performance for the two feature classification performance averaged across all ranges and noise levels for both Filter Type 1 and Filter Type 2 is shown in Table 4 and Table 5.

Table 4
Two Feature Correct Classification Percentage For Filter Type 1 Averaged Across All Range and Noise Levels

Approach	% Correct
PCA ANN	81.05
ANN Ensemble	79.03
Max Likelihood Ensemble	78.92
PCA Max Likelihood	78.27
LDA Max Likelihood	77.40
Native ANN	77.17
Euclidean Ensemble	75.62
LDA Euclidean	75.58
LDA ANN	75.53
Native Max Likelihood	73.38
PCA Euclidean	71.87
Native Euclidean	67.13

Table 5
Two Feature Correct Classification Percentage For Filter Type 2 Averaged Across All Ranges and Noise Levels

Approach	% Correct
Native ANN	100.00
PCA Max Likelihood	100.00
PCA ANN	100.00
LDA Euclidean	100.00
LDA Max Likelihood	100.00
LDA ANN	100.00
Euclidean Ensemble	100.00
Max Likelihood Ensemble	100.00
ANN Ensemble	100.00
Native Max Likelihood	99.68
PCA Euclidean	99.48
Native Euclidean	95.70

Filter Type 1, the two six state filters, did well initially, but performance dropped off significantly as range increased. This indicates the two six state filter approach is sensitive to the effects of noise, which are linearly amplified as a function of distance. Intuitively, this is consistent with the model being used to construct the vectors in 3D space. Given that the 3D location of each point of interest is being reconstructed independently, and the noise on each angle measurement for each point is independent, the predicted geometries can easily diverge in the presence of significant noise and lead to highly skewed feature values. Filter Type 2, the combined nine state filter, consistently performed well, thus demonstrating its robustness for dealing with noise and added range. This was a result of the nine state filters a priori knowledge of the rigid geometry connecting the PoI which served to constrain the geometry to its true form. This outcome was confirmed by both the plotted data and table of average results. Consequently, we conclude Filter Type 2 is the superior solution for estimating the required parameters for implementing the two feature approach.

Filter Type 2 do not provide much insight into the effects of evaluating the features in different transform domains, simply because of how well all of the classifiers did in all of the data domains. Conversely, Filter Type 1 provided more varied results for the purpose of evaluating the utility of transforming the features with LDA or PCA prior to evaluation in the classifiers. Two of the three native spatial domain approaches scored in the bottom three in terms of classification performance. Additionally, the native spatial domain approach that did not score in the bottom three was the approach which implemented the feed forward neural network for classification. As noted earlier in the text, the neural network maps the native spatial domain data into a new domain before classification occurs. Essentially, the neural network is performing a domain transform operation similar to PCA or LDA prior to

classification. Therefore, we conclude the native spatial domain features can be more effectively represented in a transform domain for the purposes of classification.

Again, Filter Type 2 did not provide much insight into the effects of evaluating the different classification approaches due to the high performance levels across the board. However, Filter Type 1 results were more varied and do provide insight into the performance of the different classification approaches. The top three performers in terms of classification for Filter Type 1 were ANN based. The top performer was an ANN based ensemble approach which utilized both the LDA and PCA domain features and then fused the results using a Bayesian Network. Consequently, the ensemble approach benefited from the advantages of using both the LDA and PCA domain features.

CHAPTER FIVE:
THREE FEATURE CLASSIFICATION OF VEHICLES

Euler angles provide a mechanism to represent orthogonal angular information with separate values [128][129][130], rather than collapsing all angular information into a single value as is done with a dot product. However, the values of the Euler angles are dependent on the reference frame to which they are relative. For the purposes of classification, one must choose a reference frame in which the Euler angles are constant. For the three feature approach detailed in this section, the reference frame to which the Euler angles are referenced will be calculated using the velocity vector of PoI A of the vehicle. The velocity reference frame is referred to as frame V. The X-Axis of the reference frame V will be aligned with the velocity vector of PoI A on the vehicle. They Y-Axis and Z-Axis will be calculated using cross product operations and when combined with the X-Axis will result in a right handed coordinate system to serve as reference frame V. The velocity vector of PoI A is assumed to have a constant orientation to the vector between PoI A and PoI B on the vehicle regardless of vehicle orientation. It's worth re-noting, one of the requirements for the proposed techniques is the velocity vector of the vehicle is constant. In other words, the vehicle can't be turning or changing lanes. Given that the velocity vector is being used a reference frame, the three feature approach is only applicable to moving vehicles. Equation 242 through Equation 244 demonstrates the calculation of the features for the three feature classification approach. The features are derived either from the Euler angles or the lengths of the vectors.

$$\theta^{V}_{AB_i} = \sin^{-1}\left(\vec{U}^{V}_{AB_{z_i}}\right) \quad (242)$$

$$\psi^{V}_{AB_i} = \tan^{-1}\left(\frac{\vec{X}^{V}_{AB_{y_i}}}{\vec{X}^{V}_{AB_{x_i}}}\right) \quad (243)$$

$$r_{AB_i} = \sqrt{\left(\vec{X}^{R}_{B} - \vec{X}^{R}_{A}\right)^2} \quad (244)$$

The three feature classification approach was implemented using two separate Kalman Filter formulations. The first formulation, referred to as Filter Type 1, employed two separate six state filters which each independently tracked a point of interest. Using the 3D position information contained in each six state filter for their respective point of interest, the vectors and angles of interest were calculated. The second formulation, referred to as Filter Type 2, employed a single nine state filter which use knowledge of the feature geometry to track both PoI as a single rigid structure. Using the 3D position information for the PoI contained in the nine state filter, the vectors and angles of interest were calculated. The symbols corresponding to the different domain data types tested in each classifier are depicted in Figure 40.

Data Domain	Symbol
Native Spatial	
PCA	
LDA	
Ensemble	

Figure 40 Data Domain Three Feature Classification Plot Symbols

Experimental Results – Filter Type 1 - Six State Filter

The classification performance of Filter Type 1 for the three feature approach is detailed in this section for constant velocity vehicles with varying levels of noise.

Noise Level of 50 microradians

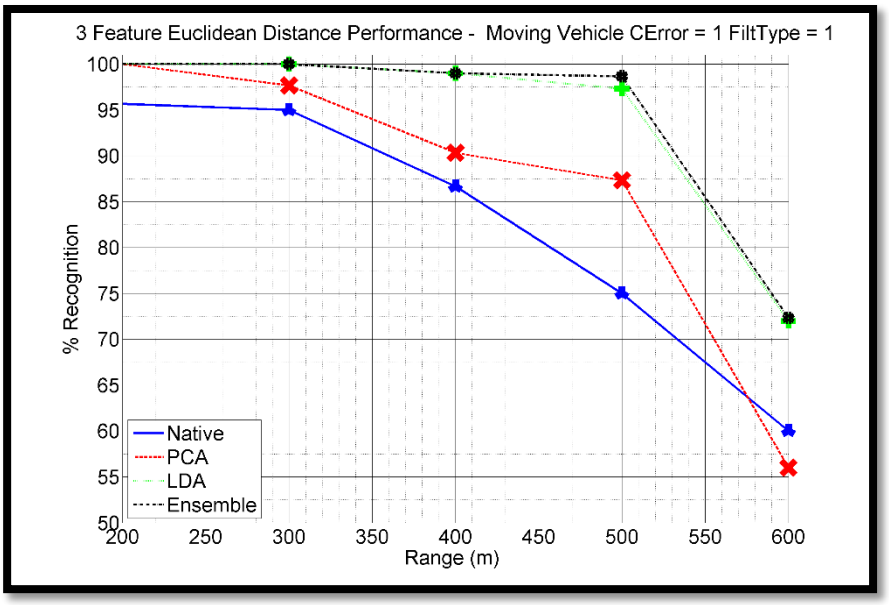

Figure 41 Percent Recognition of Vehicle Types -vs- Range with Three Feature Euclidean Distance: Moving Vehicle - Noise Level of 50 microradians - Filter Type 1

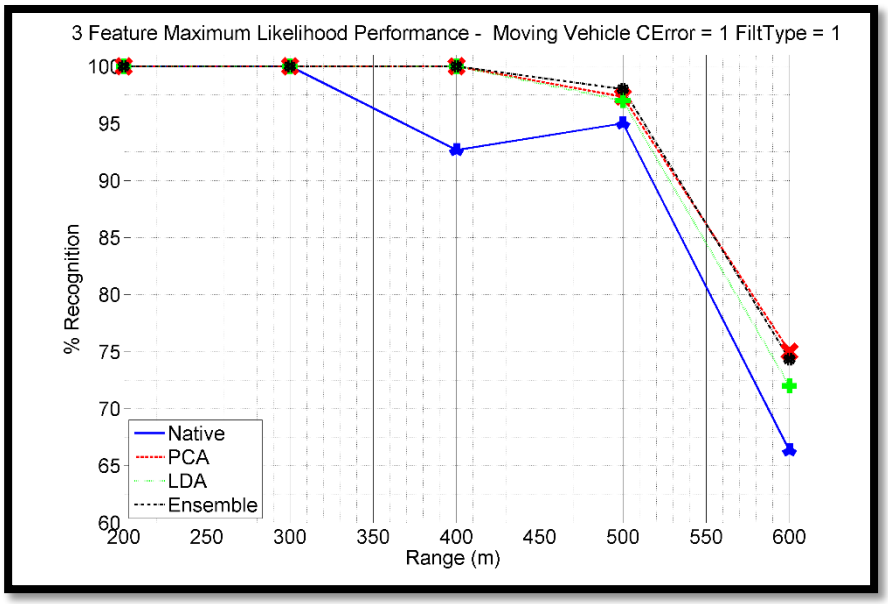

Figure 42 Percent Recognition of Vehicle Types -vs- Range with Three Feature Max Likelihood: Moving Vehicle - Noise Level of 50 microradians – Filter Type 1

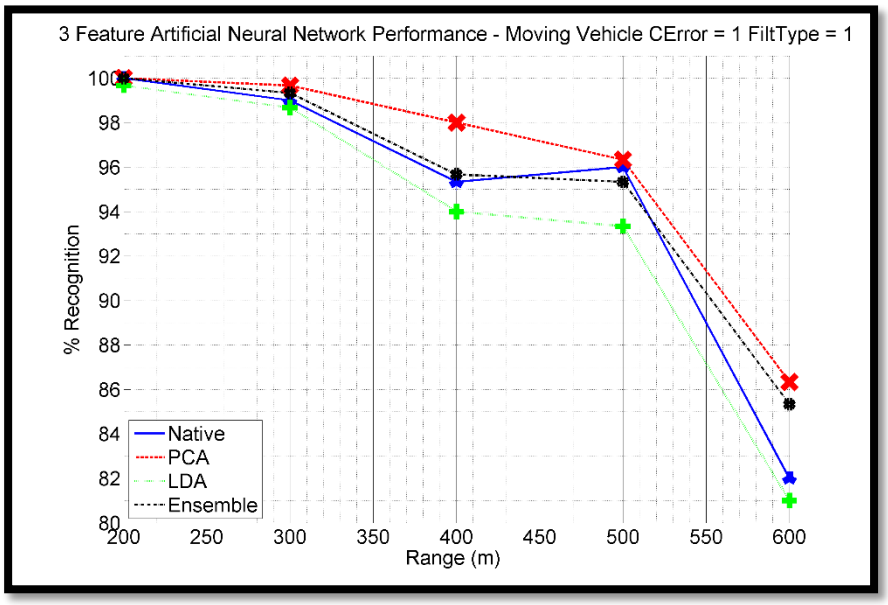

Figure 43 Percent Recognition of Vehicle Types -vs- Range with Three Feature ANN: Moving Vehicle - Noise Level of 50 microradians - Filter Type 1

Noise Level of 100 microradians

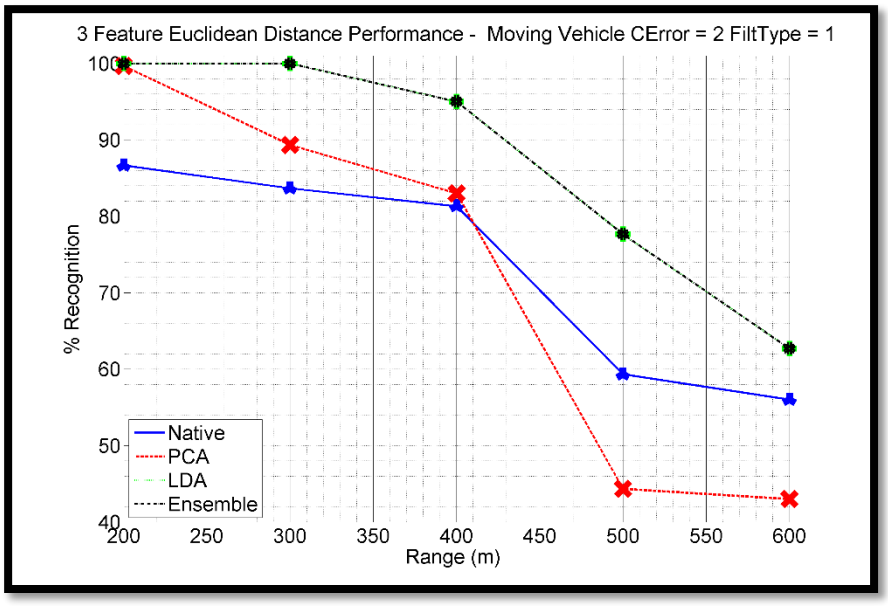

Figure 44 Percent Recognition of Vehicle Types -vs- Range with Three Feature Euclidean Distance: Moving Vehicle - Noise Level of 100 microradians - Filter Type 1

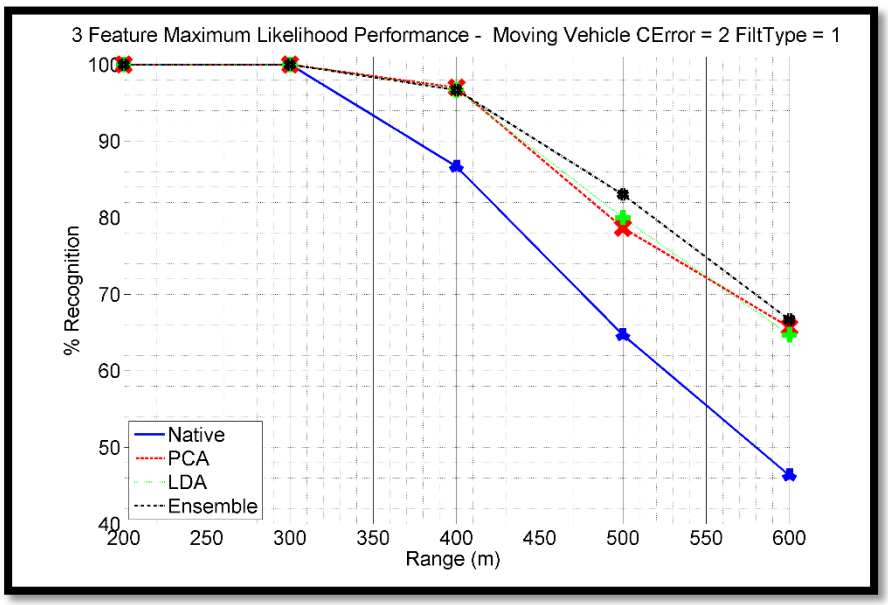

Figure 45 Percent Recognition of Vehicle Types -vs- Range with Three Feature Max Likelihood: Moving Vehicle - Noise Level of 100 microradians – Filter Type 1

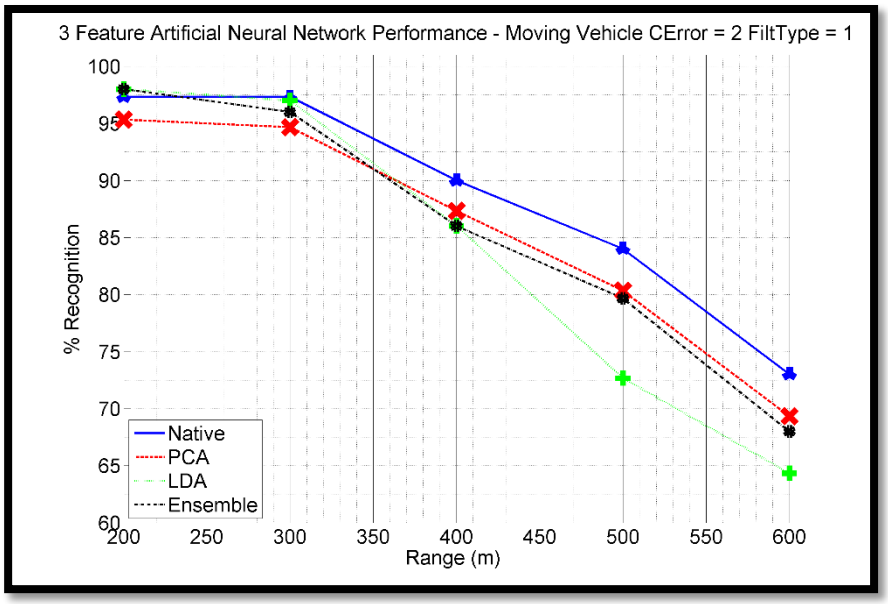

Figure 46 Percent Recognition of Vehicle Types -vs- Range with Three Feature ANN: Moving Vehicle - Noise Level of 100 microradians - Filter Type 1

Noise Level of 150 microradians

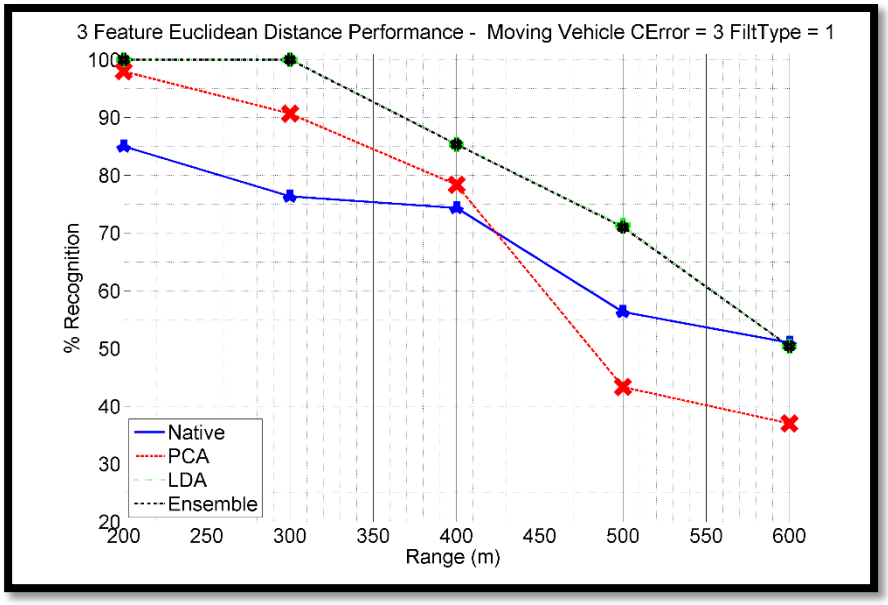

Figure 47 Percent Recognition of Vehicle Types -vs- Range with Three Feature Euclidean Distance: Moving Vehicle - Noise Level of 150 microradians - Filter Type 1

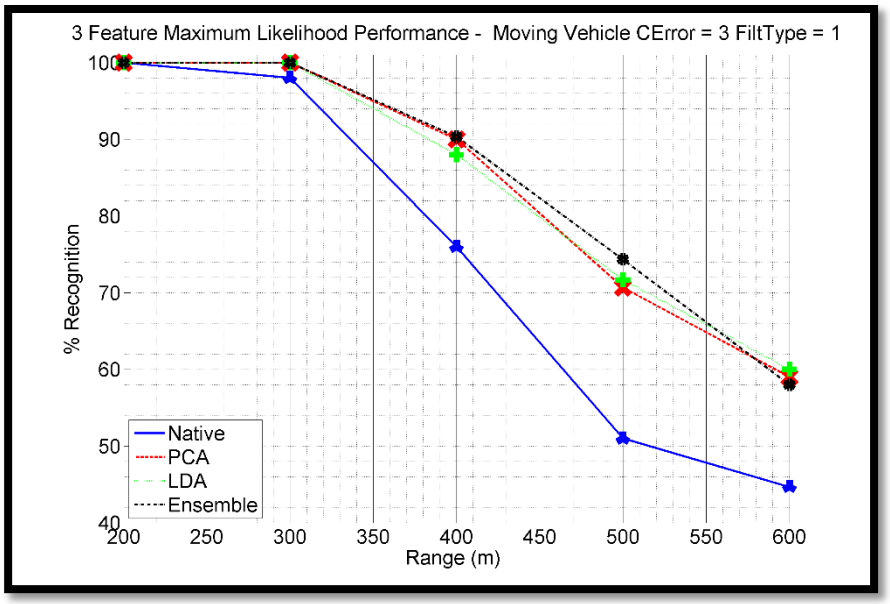

Figure 48 Percent Recognition of Vehicle Types -vs- Range with Three Feature Max Likelihood: Moving Vehicle - Noise Level of 150 microradians – Filter Type 1

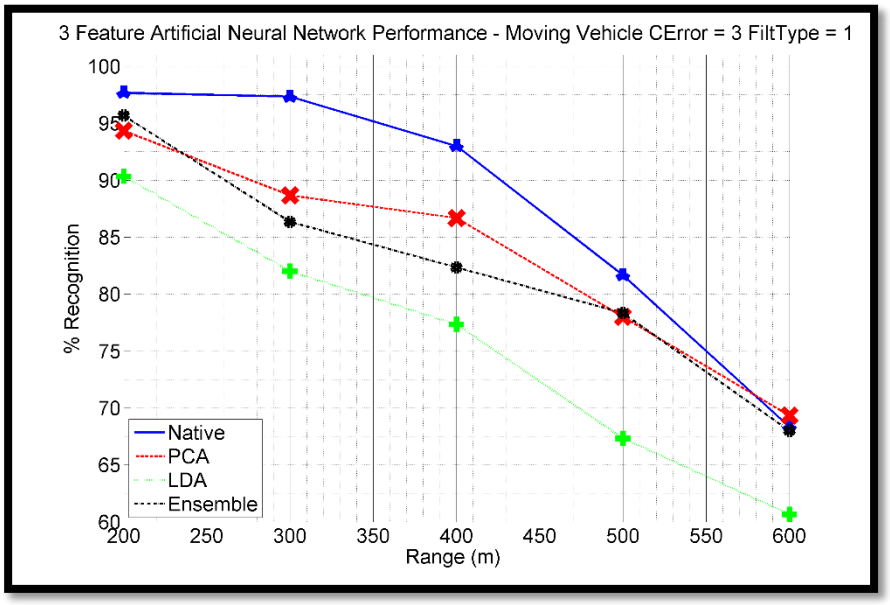

Figure 49 Percent Recognition of Vehicle Types -vs- Range with Three Feature ANN: Moving Vehicle - Noise Level of 150 microradians - Filter Type 1

Noise Level of 200 microradians

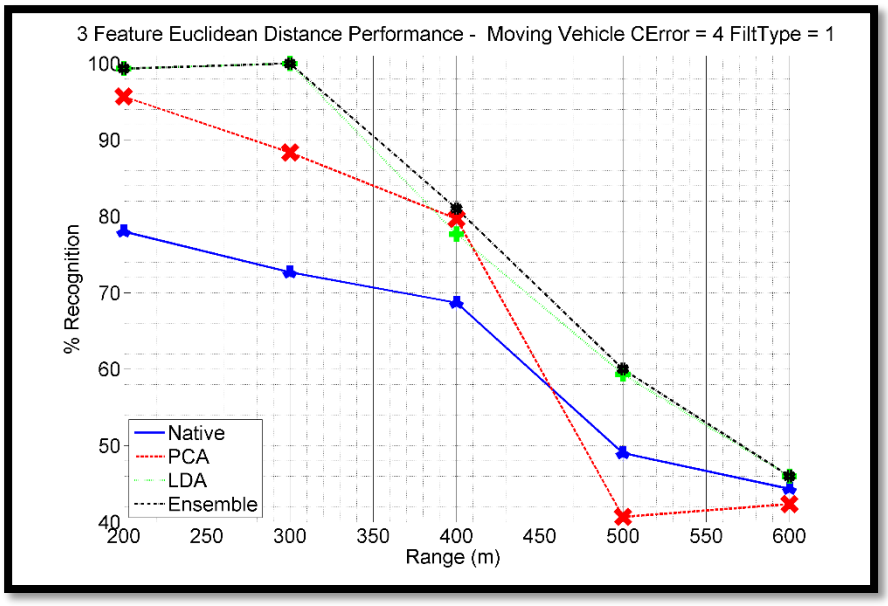

Figure 50 Percent Recognition of Vehicle Types -vs- Range with Three Feature Euclidean Distance: Moving Vehicle - Noise Level of 200 microradians - Filter Type 1

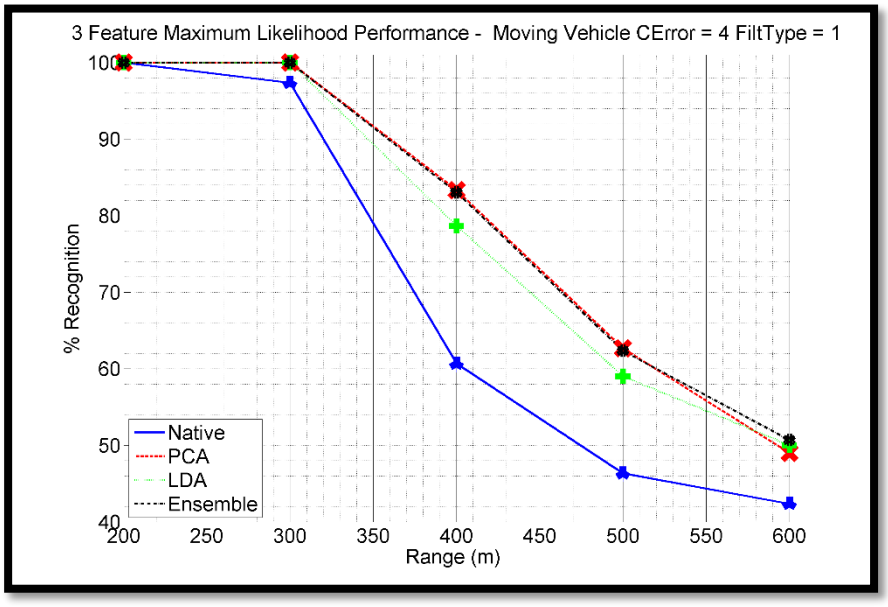

Figure 51 Percent Recognition of Vehicle Types -vs- Range with Three Feature Max Likelihood: Moving Vehicle - Noise Level of 200 microradians – Filter Type 1

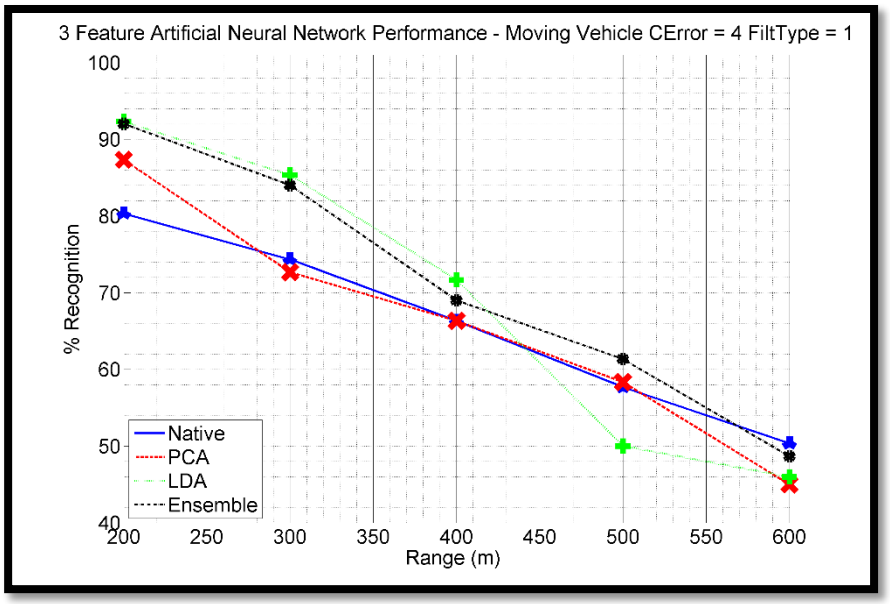

Figure 52 Percent Recognition of Vehicle Types -vs- Range with Three Feature ANN: Moving Vehicle - Noise Level of 200 microradians - Filter Type 1

Experimental Results – Filter Type 2 - Nine State Filter

The classification performance of Filter Type 2 for the three feature approach is detailed in this section for constant velocity vehicles with varying levels of noise.

Noise Level of 50 microradians

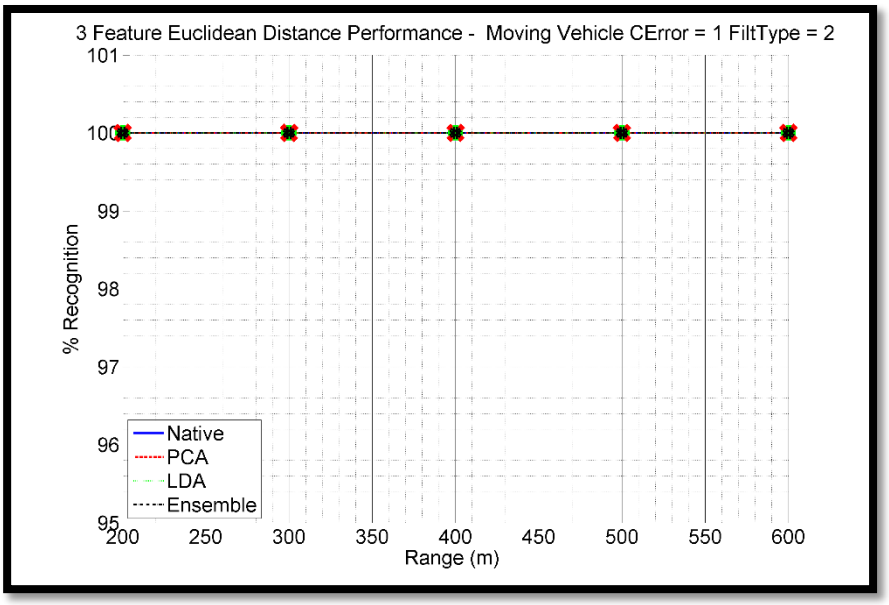

Figure 53 Percent Recognition of Vehicle Types -vs- Range with Three Feature Euclidean Distance: Moving Vehicle - Noise Level of 50 microradians - Filter Type 2

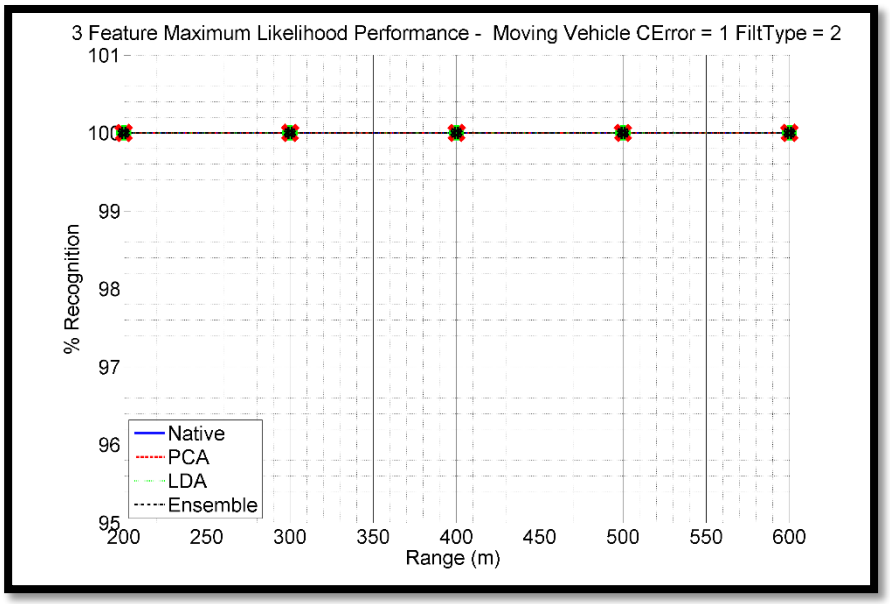

Figure 54 Percent Recognition of Vehicle Types -vs- Range with Three Feature Max Likelihood: Moving Vehicle - Noise Level of 50 microradians – Filter Type 2

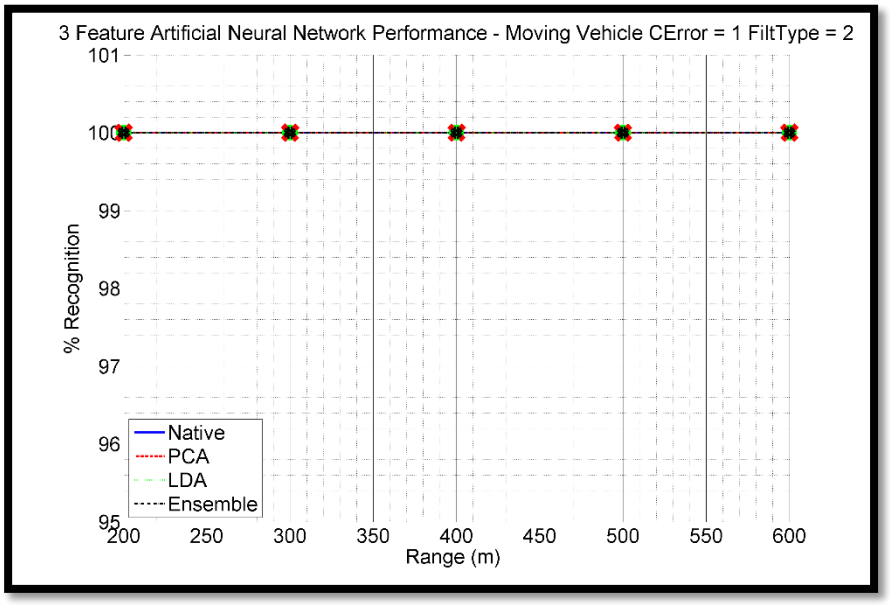

Figure 55 Percent Recognition of Vehicle Types -vs- Range with Three Feature ANN: Moving Vehicle - Noise Level of 50 microradians - Filter Type 2

Noise Level of 100 microradians

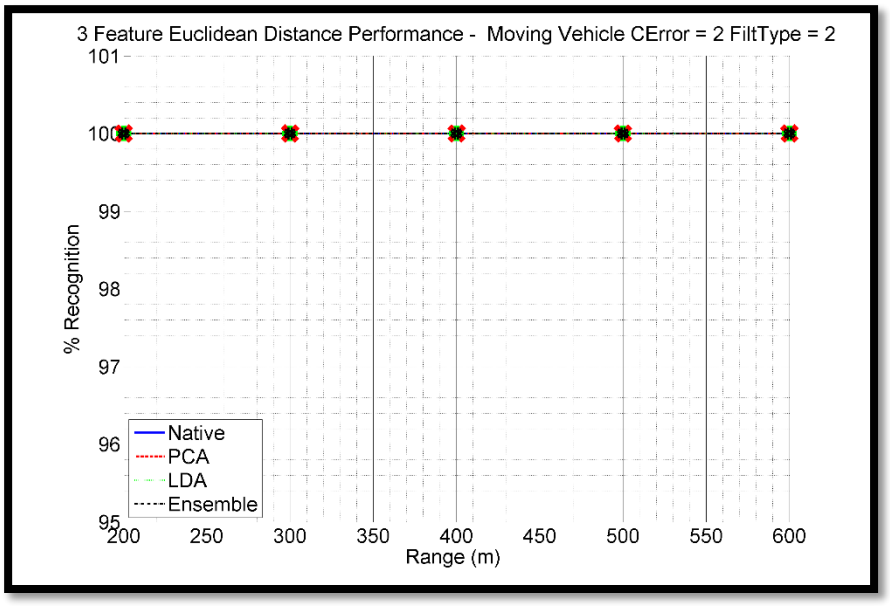

Figure 56 Percent Recognition of Vehicle Types -vs- Range with Three Feature Euclidean Distance: Moving Vehicle - Noise Level of 100 microradians - Filter Type 2

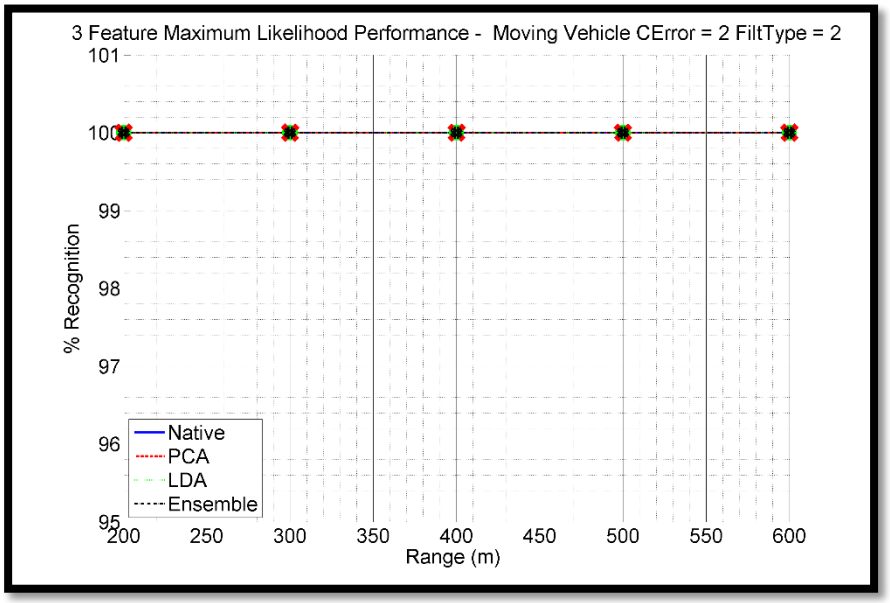

Figure 57 Percent Recognition of Vehicle Types -vs- Range with Three Feature Max Likelihood: Moving Vehicle - Noise Level of 100 microradians – Filter Type 2

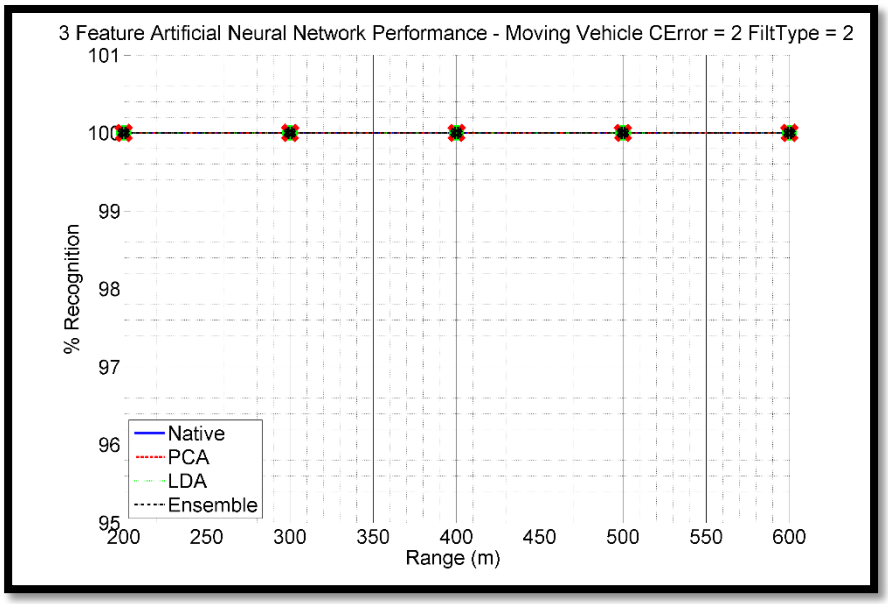

Figure 58 Percent Recognition of Vehicle Types -vs- Range with Three Feature ANN: Moving Vehicle - Noise Level of 100 microradians - Filter Type 2

Noise Level of 150 microradians

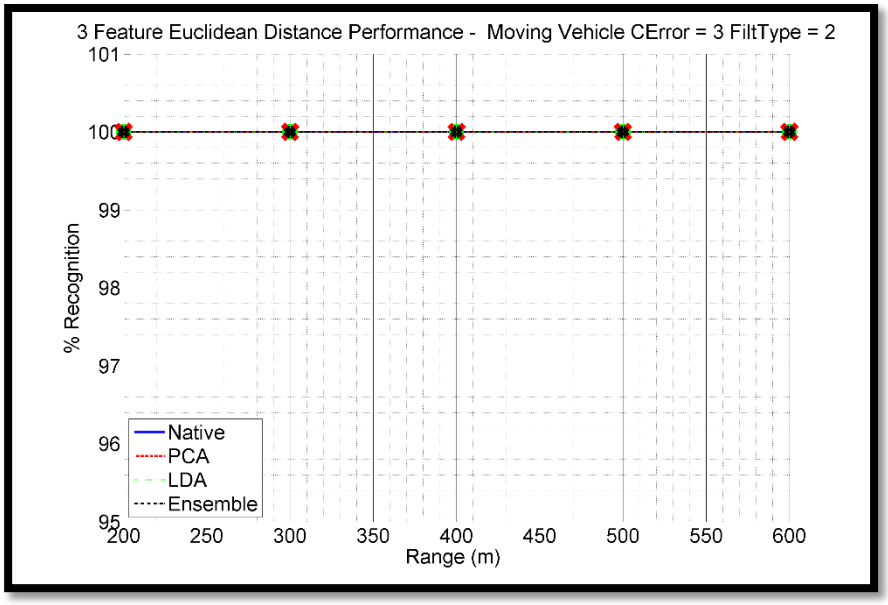

Figure 59 Percent Recognition of Vehicle Types -vs- Range with Three Feature Euclidean Distance: Moving Vehicle - Noise Level of 150 microradians - Filter Type 2

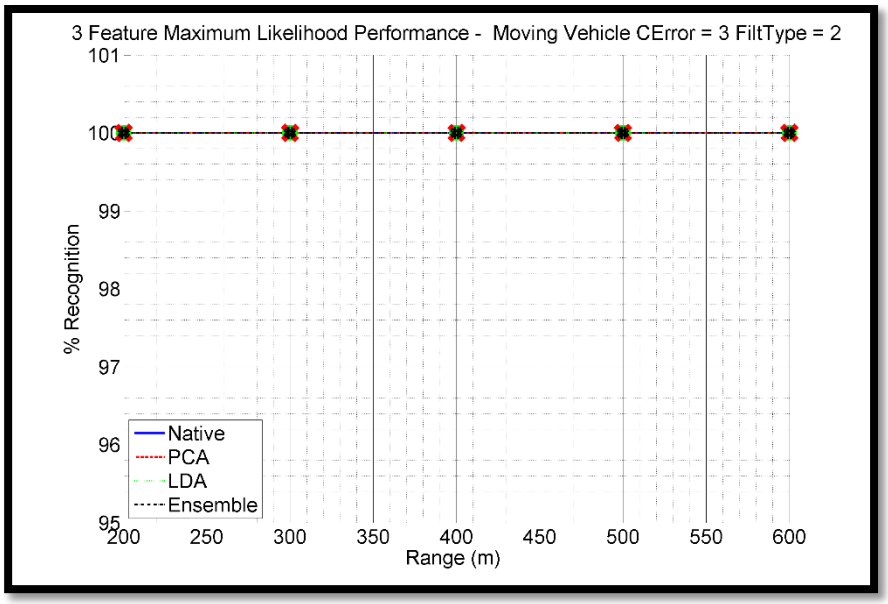

Figure 60 Percent Recognition of Vehicle Types -vs- Range with Three Feature Max Likelihood: Moving Vehicle - Noise Level of 150 microradians – Filter Type 2

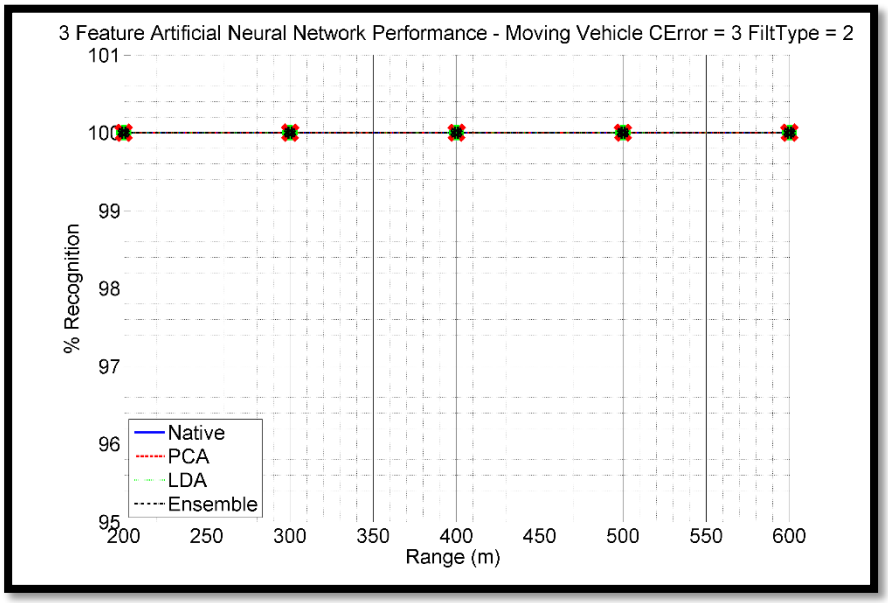

Figure 61 Percent Recognition of Vehicle Types -vs- Range with Three Feature ANN: Moving Vehicle - Noise Level of 150 microradians - Filter Type 2

Noise Level of 200 microradians

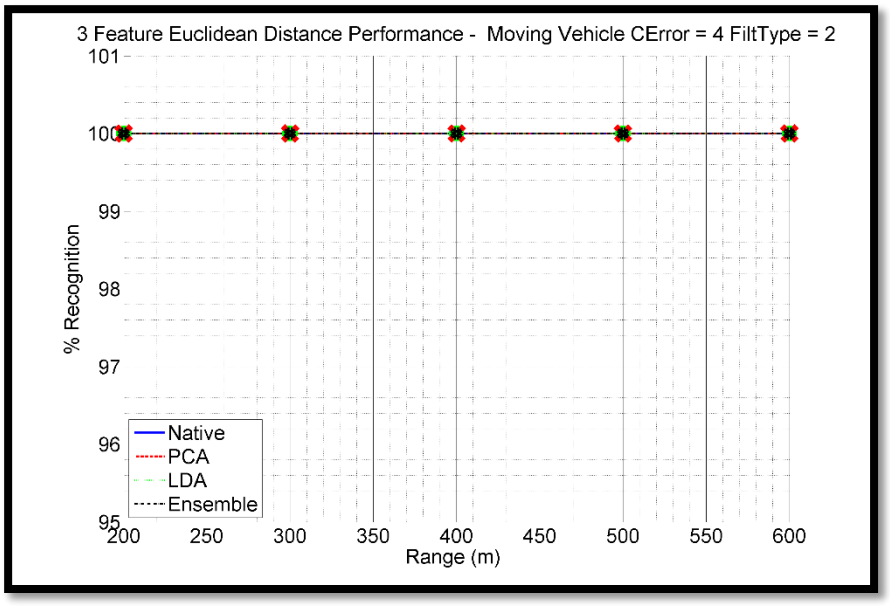

Figure 62 Percent Recognition of Vehicle Types -vs- Range with Three Feature Euclidean Distance: Moving Vehicle - Noise Level of 200 microradians - Filter Type 2

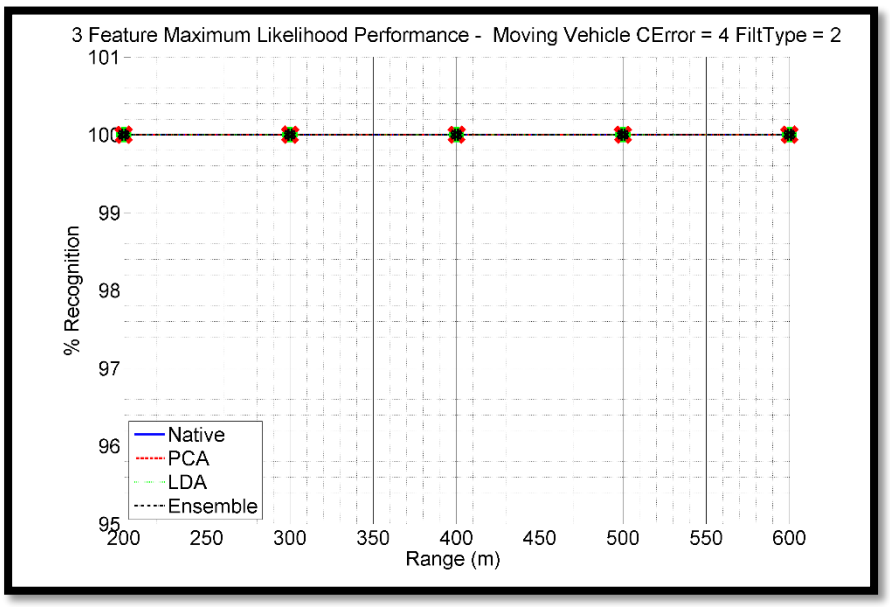

Figure 63 Percent Recognition of Vehicle Types -vs- Range with Three Feature Max Likelihood: Moving Vehicle - Noise Level of 200 microradians – Filter Type 2

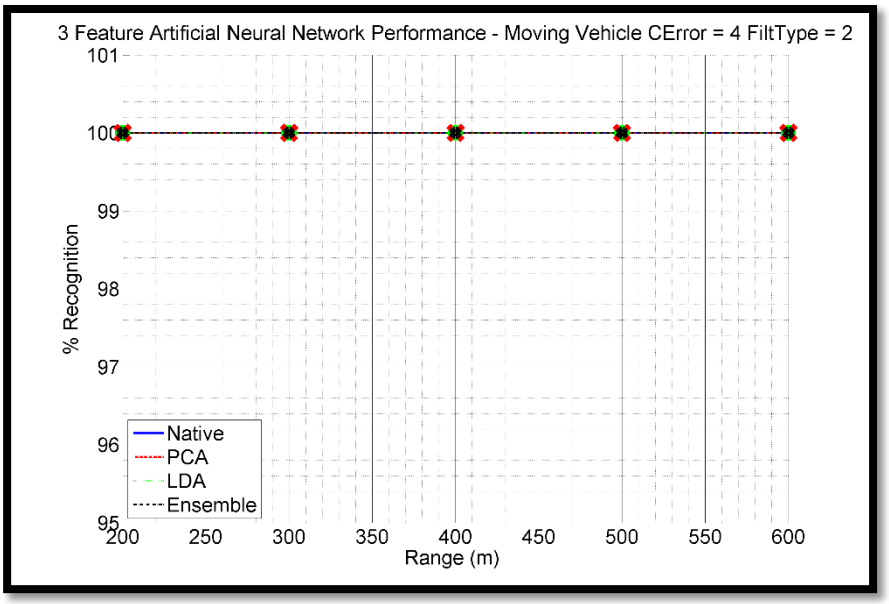

Figure 64 Percent Recognition of Vehicle Types -vs- Range with Three Feature ANN: Moving Vehicle - Noise Level of 200 microradians - Filter Type 2

Three Feature Conclusions

The performance for the three feature classification performance averaged across all ranges and noise levels for both Filter Type 1 and Filter Type 2 is shown in Table 6 and Table 7.

Table 6
Three Feature Correct Classification Percentage For Filter Type 1 Averaged Across All Range and Noise Levels

Approach	% Correct
Max Likelihood Ensemble	86.87
PCA Max Likelihood	86.42
LDA Max Likelihood	85.88
Euclidean Ensemble	84.92
LDA Euclidean	84.65
Native ANN	84.05
ANN Ensemble	83.45
PCA ANN	82.70
LDA ANN	80.38
Native Max Likelihood	78.40
PCA Euclidean	74.23
Native Euclidean	71.75

Table 7
Three Feature Correct Classification Percentage For Filter Type 2 Averaged Across All Ranges and Noise Levels

Approach	% Correct
Native ANN	100.00
PCA Max Likelihood	100.00
PCA ANN	100.00
LDA Euclidean	100.00
LDA Max Likelihood	100.00
LDA ANN	100.00
Euclidean Ensemble	100.00
Max Likelihood Ensemble	100.00
ANN Ensemble	100.00
Native Max Likelihood	100.00
PCA Euclidean	100.00
Native Euclidean	100.00

Filter Type 1, the two six state filters, did well initially, but performance dropped off significantly as range increased. This indicates the two six state filter approach is sensitive to the effects of noise, which are linearly amplified as a function of distance. Intuitively, this is consistent with the model being used to construct the vectors in 3D space. Given that the 3D location of each point of interest is being reconstructed independently, and the noise on each angle measurement for each point is independent, the predicted geometries can easily diverge in the presence of significant noise and lead to highly skewed feature values. Filter Type 2, the combined nine state filter, consistently performed well, thus demonstrating its robustness for dealing with noise and added range. This was a result of the combined nine state filters a priori knowledge of the rigid geometry connecting the PoI which served to constrain the geometry to its true form. This outcome was confirmed by both the plotted data and table of average results. Consequently, we conclude Filter Type 2 is the superior solution for estimating the required parameters to implement the three feature approach.

Filter Type 2 results do not provide much insight into the effects of evaluating the features in different transform domains, simply because of how well all of the classifiers did in all of the data domains Conversely, Filter Type 1 provides more varied results for the purpose of evaluating the utility of transforming the features with LDA or PCA prior to evaluation in the classifiers. Two of the three native spatial domain approaches scored in the bottom three in terms of classification performance. Additionally, the native spatial domain approach that did not score in the bottom three was the approach which implemented the feed forward neural network for classification. As noted earlier in the text, the neural network maps the native spatial domain data into a new domain before classification occurs. Essentially, the neural network is performing a domain transform operation similar to PCA

or LDA prior to classification. Therefore, we conclude the native spatial domain features can be more effectively represented in a transform domain resulting in superior classification performance.

Again, Filter Type 2 did not provide much insight into the effects of evaluating the different classification approaches due to the high performance levels across the board. However, Filter Type 1 results were more varied and do provide insight into the performance of the different classification approaches. The top four performers using Filter Type 1 were transform domain based classifiers. The top performer was the Maximum Likelihood ensemble approach which implemented a Maximum Likelihood classifier for both the LDA and PCA domain features, and then fused the results of the independent classifiers into a single result using a Bayesian Network. Consequently, the ensemble approach benefited from the advantages of using both the LDA and PCA domain features.

CHAPTER SIX:
SEVEN FEATURE CLASSIFICATION OF VEHICLES

Euler angles provide a mechanism to represent orthogonal angular information with separate values [128][129][130], rather than collapsing all angular information into a single value as is done with a dot product. However, the values of the Euler angles are dependent on the reference frame to which they are relative. For the purposes of classification, one must choose a reference frame in which the Euler angles are constant. For the seven feature approach detailed in this section, the reference frame to which the Euler angles are referenced will be calculated using the vector between PoI *A* and PoI *B* on the vehicle. The reference frame is referred to as frame *AB*. The X-Axis of the reference frame *AB* will be aligned with the *AB* vector of the vehicle. They Y-Axis and Z-Axis will be calculated using cross product operations and when combined with the X-Axis will result in a right handed coordinate system to serve as reference frame *AB*. Equation 245 through Equation 251 demonstrates the calculation of the features for the seven feature classification approach.

$$\boldsymbol{r_{AB_i}} = \sqrt{\left(\vec{\boldsymbol{X}}_{\boldsymbol{B}}^{\boldsymbol{R}} - \vec{\boldsymbol{X}}_{\boldsymbol{A}}^{\boldsymbol{R}}\right)^2} \tag{245}$$

$$\boldsymbol{r_{AC_i}} = \sqrt{\left(\vec{\boldsymbol{X}}_{\boldsymbol{C}}^{\boldsymbol{R}} - \vec{\boldsymbol{X}}_{\boldsymbol{A}}^{\boldsymbol{R}}\right)^2} \tag{246}$$

$$\boldsymbol{r_{BC_i}} = \sqrt{\left(\vec{\boldsymbol{X}}_{\boldsymbol{B}}^{\boldsymbol{R}} - \vec{\boldsymbol{X}}_{\boldsymbol{C}}^{\boldsymbol{R}}\right)^2} \tag{247}$$

$$\boldsymbol{\theta_{AC_i}^{AB} = \sin^{-1}\left(\vec{U}_{AC_{z_i}}^{AB}\right)} \quad (248)$$

$$\boldsymbol{\psi_{AC_i}^{AB} = \tan^{-1}\left(\frac{\vec{X}_{AC_{y_i}}^{AB}}{\vec{X}_{AC_{x_i}}^{AB}}\right)} \quad (249)$$

$$\boldsymbol{\theta_{BC_i}^{AB} = \sin^{-1}\left(\vec{U}_{BC_{z_i}}^{AB}\right)} \quad (250)$$

$$\boldsymbol{\psi_{BC_i}^{AB} = \tan^{-1}\left(\frac{\vec{X}_{BC_{y_i}}^{AB}}{\vec{X}_{BC_{x_i}}^{AB}}\right)} \quad (251)$$

The seven feature classification approach was implemented using two separate Kalman Filter formulations. The first formulation, referred to as Filter Type 1, employed three separate six state filters which each independently tracked a PoI. Using the 3D position information contained in each six state filter for their respective PoI, the vectors and angles of interest were calculated. The second formulation, referred to as Filter Type 3, employed a single twelve state filter which use knowledge of the feature geometry to track all three PoI as a single rigid structure. Using the 3D position information for the PoI contained in the twelve state filter, the vectors and angles of interest were calculated. The symbols corresponding to the different domain data types tested in each classifier are depicted in Figure 65.

Data Domain	Symbol
Native Spatial	
PCA	
LDA	
Ensemble	

Figure 65 Data Domain Seven Feature Classification Plot Symbols

Experimental Results – Filter Type 1 - Six State Filter

The classification performance of Filter Type 1 for the seven feature approach is detailed in this section for constant velocity vehicles with varying levels of noise.

Noise Level of 50 microradians

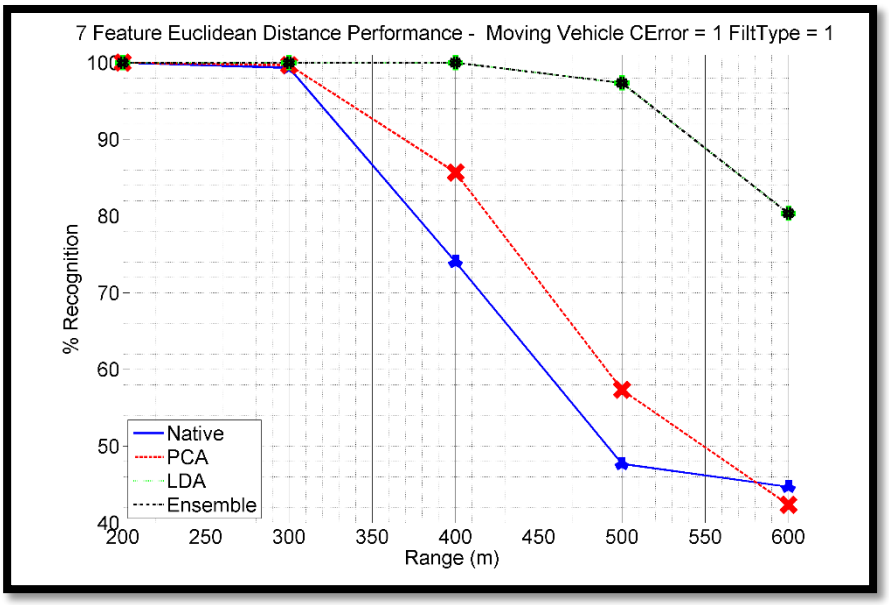

Figure 66 Percent Recognition of Vehicle Types -vs- Range with Seven Feature Euclidean Distance: Moving Vehicle - Noise Level of 50 microradians - Filter Type 1

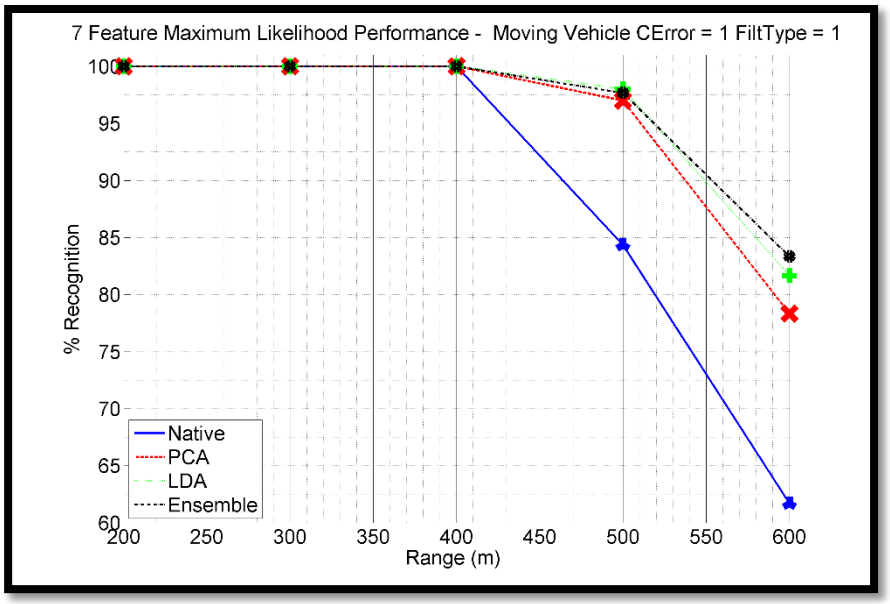

Figure 67 Percent Recognition of Vehicle Types -vs- Range with Seven Feature Max Likelihood: Moving Vehicle - Noise Level of 50 microradians – Filter Type 1

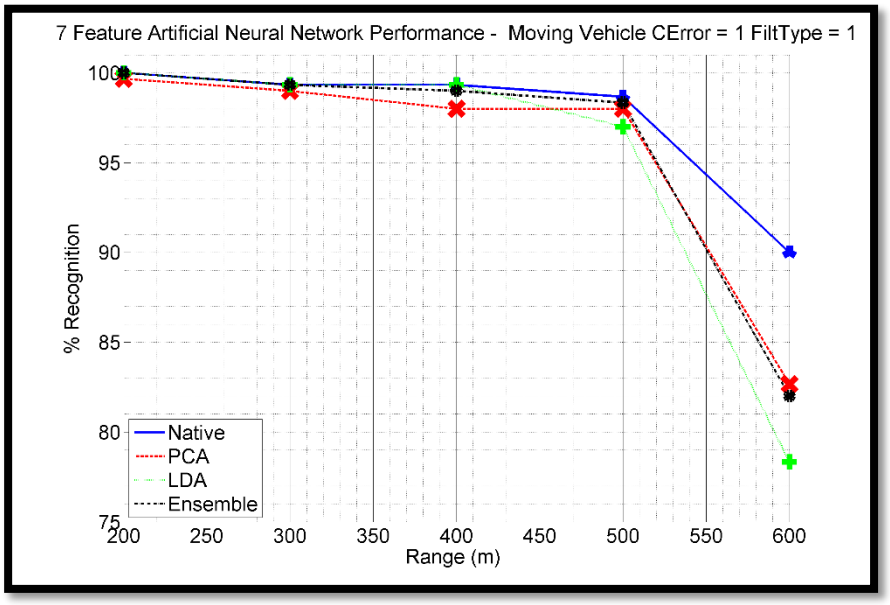

Figure 68 Percent Recognition of Vehicle Types -vs- Range with Seven Feature ANN: Moving Vehicle - Noise Level of 50 microradians - Filter Type 1

Noise Level of 100 microradians

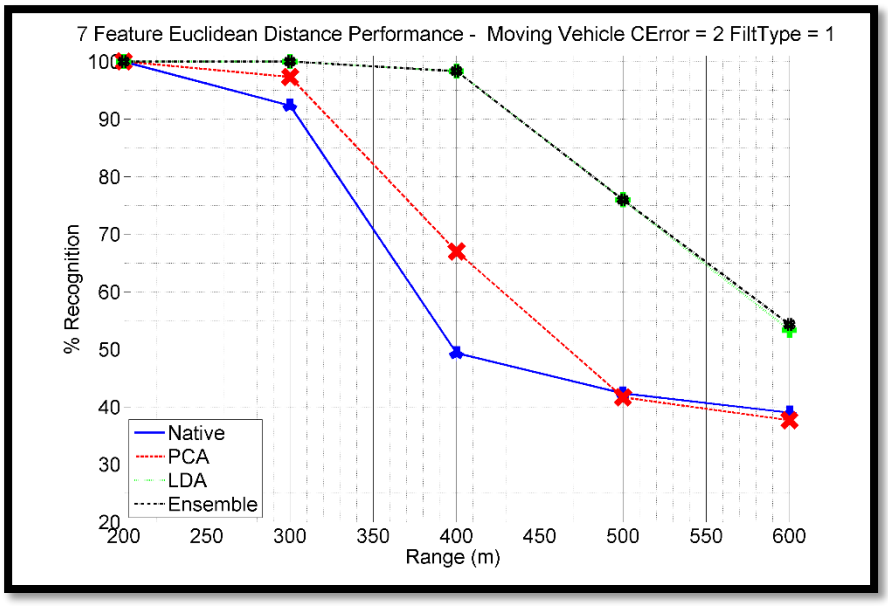

Figure 69 Percent Recognition of Vehicle Types -vs- Range with Seven Feature Euclidean Distance: Moving Vehicle - Noise Level of 100 microradians - Filter Type 1

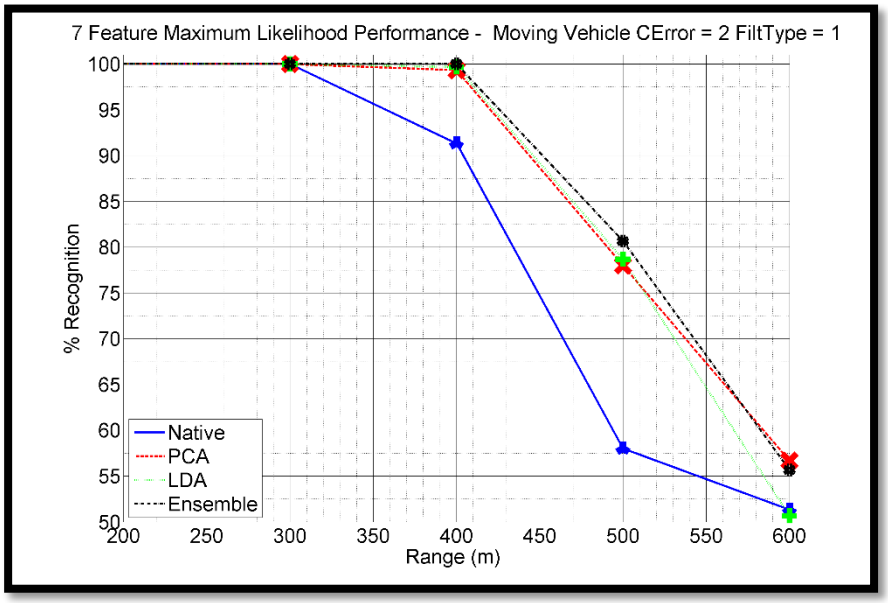

Figure 70 Percent Recognition of Vehicle Types -vs- Range with Seven Feature Max Likelihood: Moving Vehicle - Noise Level of 100 microradians – Filter Type 1

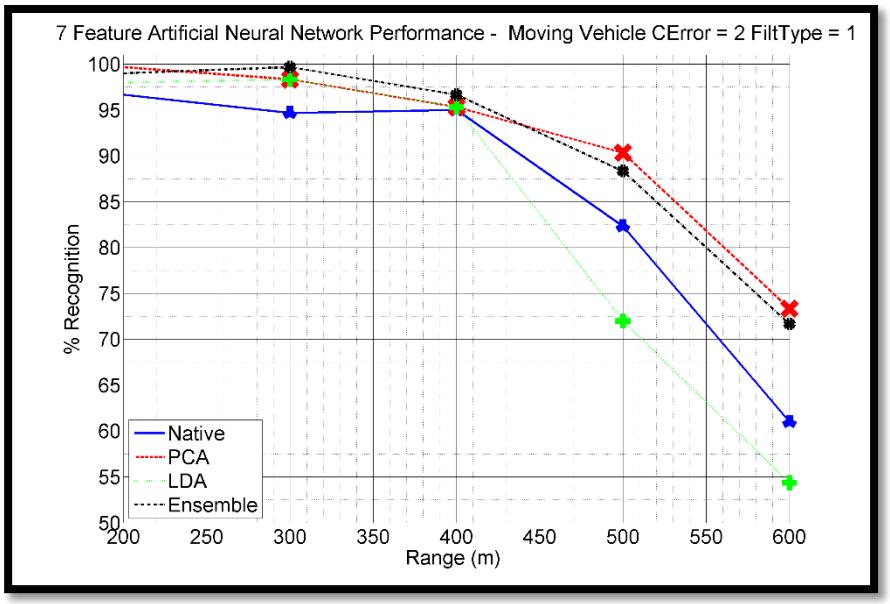

Figure 71 Percent Recognition of Vehicle Types -vs- Range with Seven Feature ANN: Moving Vehicle - Noise Level of 100 microradians - Filter Type 1

Noise Level of 150 microradians

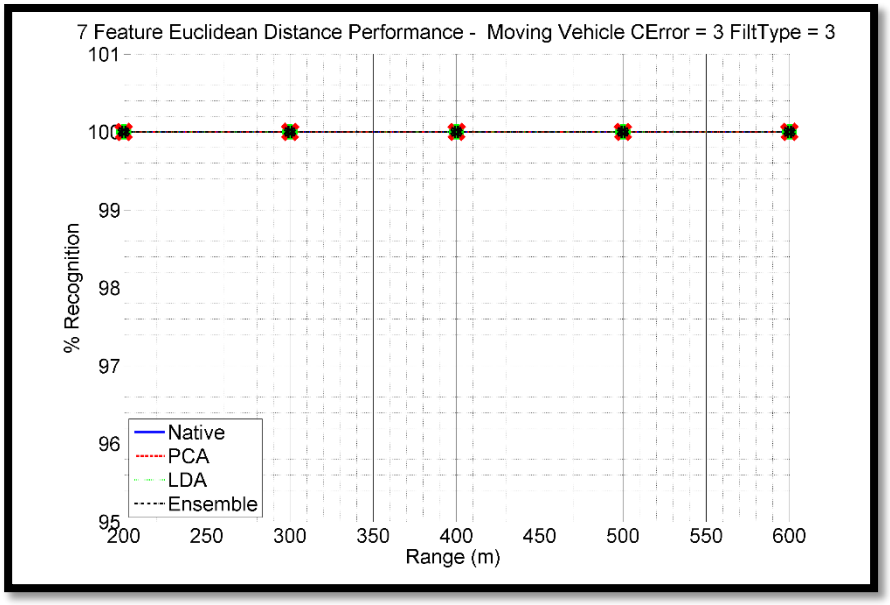

Figure 72 Percent Recognition of Vehicle Types -vs- Range with Seven Feature Euclidean Distance: Moving Vehicle - Noise Level of 150 microradians - Filter Type 1

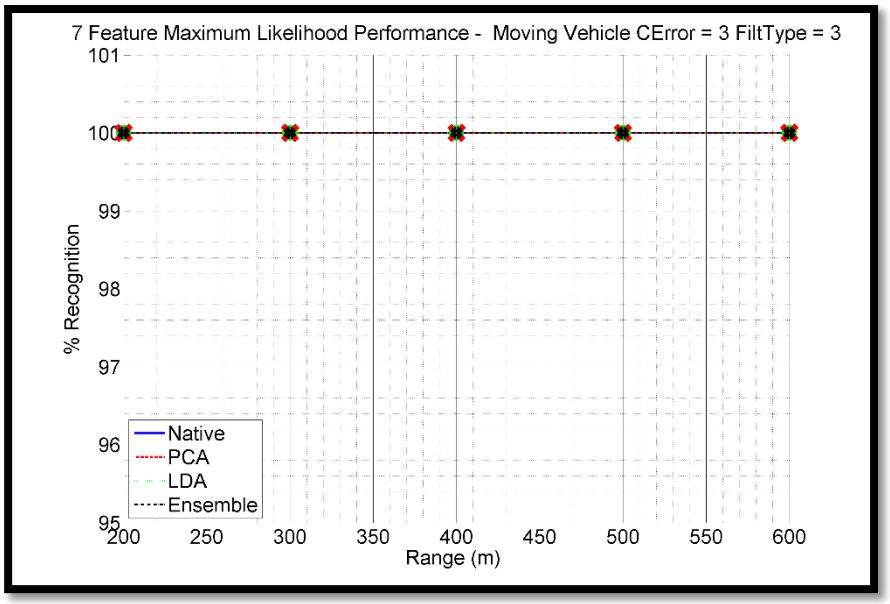

Figure 73 Percent Recognition of Vehicle Types -vs- Range with Seven Feature Max Likelihood: Moving Vehicle - Noise Level of 150 microradians – Filter Type 1

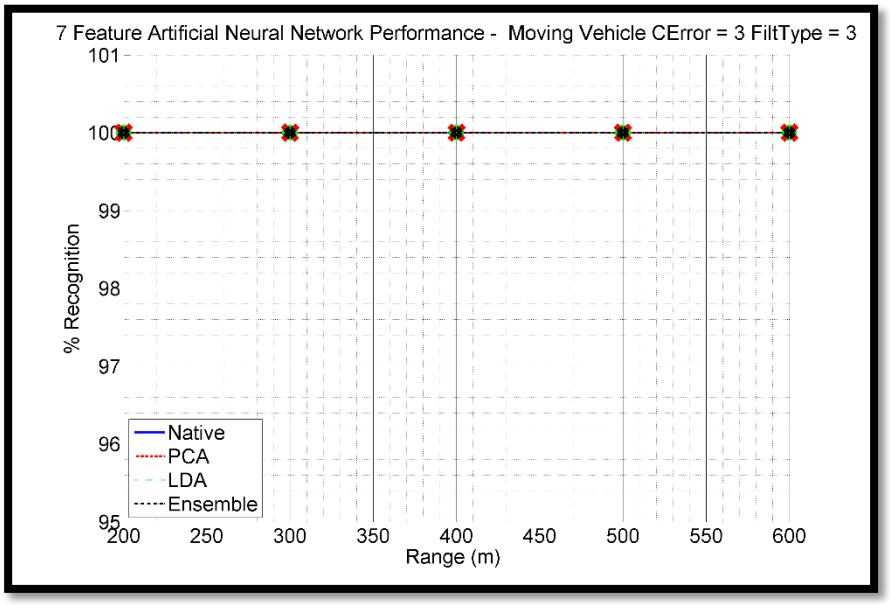

Figure 74 Percent Recognition of Vehicle Types -vs- Range with Seven Feature ANN: Moving Vehicle - Noise Level of 150 microradians - Filter Type 1

Noise Level of 200 microradians

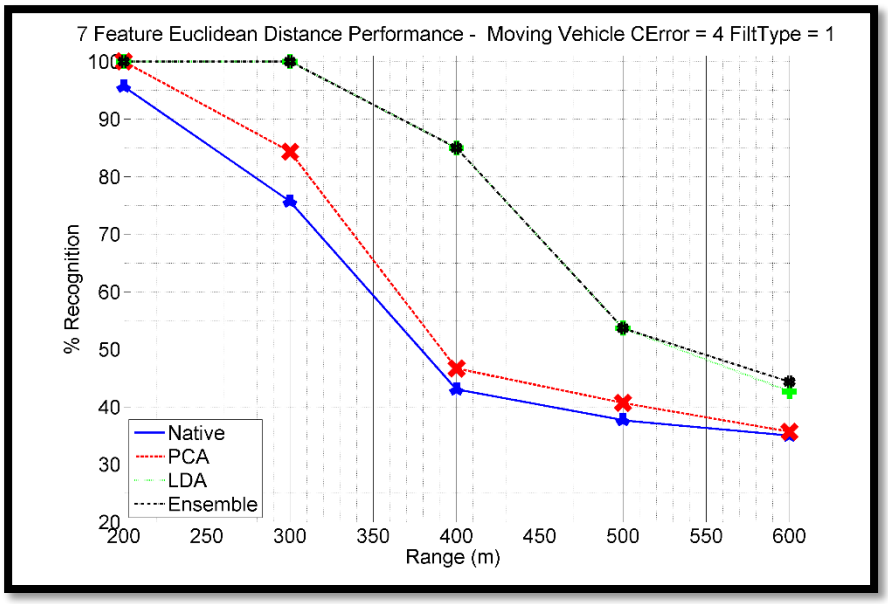

Figure 75 Percent Recognition of Vehicle Types -vs- Range with Seven Feature Euclidean Distance: Moving Vehicle - Noise Level of 200 microradians - Filter Type 1

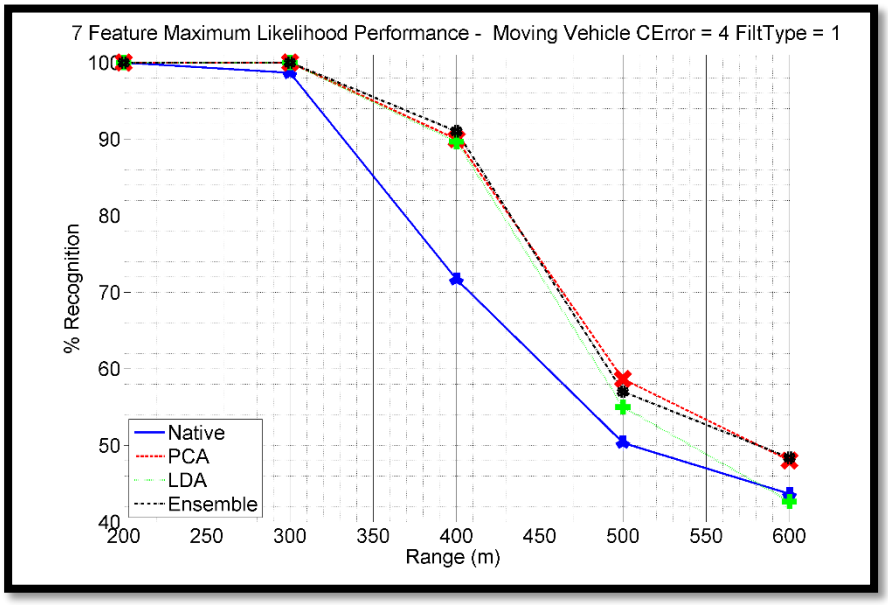

Figure 76 Percent Recognition of Vehicle Types -vs- Range with Seven Feature Max Likelihood: Moving Vehicle - Noise Level of 200 microradians – Filter Type 1

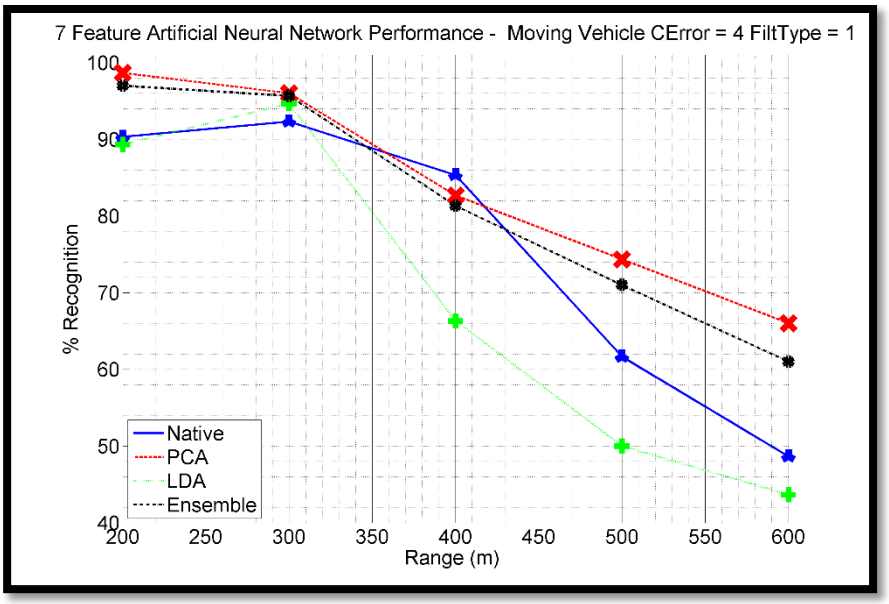

Figure 77 Percent Recognition of Vehicle Types -vs- Range with Seven Feature ANN: Moving Vehicle - Noise Level of 200 microradians - Filter Type 1

Experimental Results – Filter Type 3 - Twelve State Filter

The classification performance of Filter Type 3 for the seven feature approach is detailed in this section for constant velocity vehicles with varying levels of noise.

Noise Level of 50 microradians

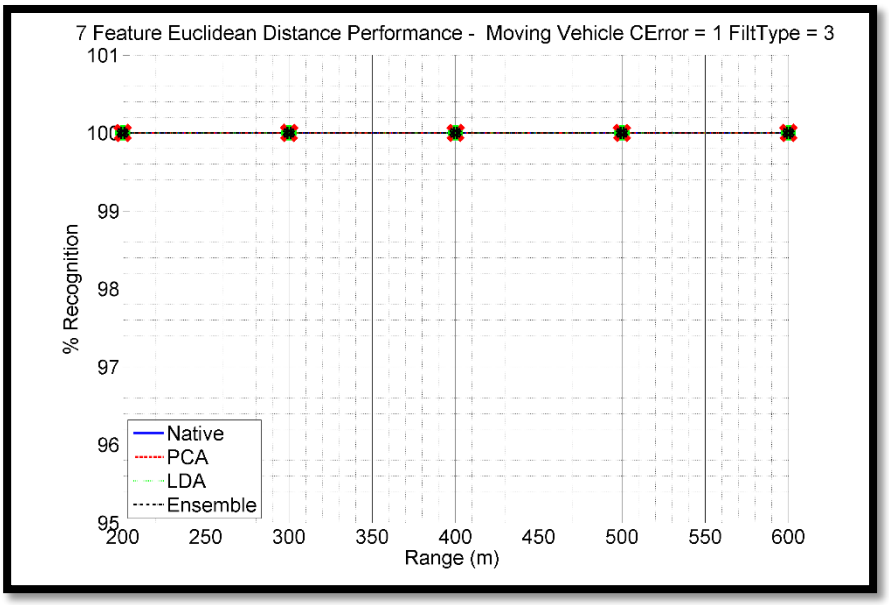

Figure 78 Percent Recognition of Vehicle Types -vs- Range with Seven Feature Euclidean Distance: Moving Vehicle - Noise Level of 50 microradians - Filter Type 3

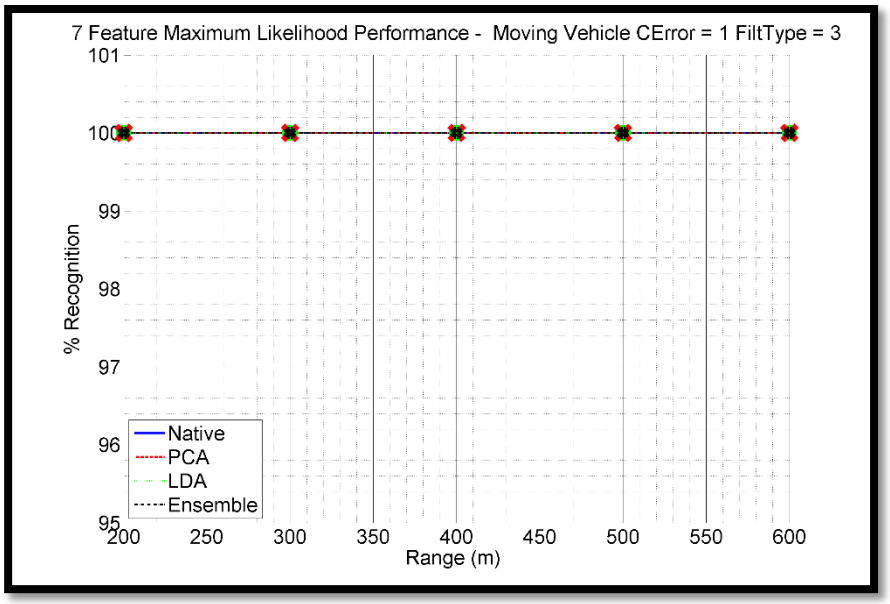

Figure 79 Percent Recognition of Vehicle Types -vs- Range with Seven Feature Max Likelihood: Moving Vehicle - Noise Level of 50 microradians – Filter Type 3

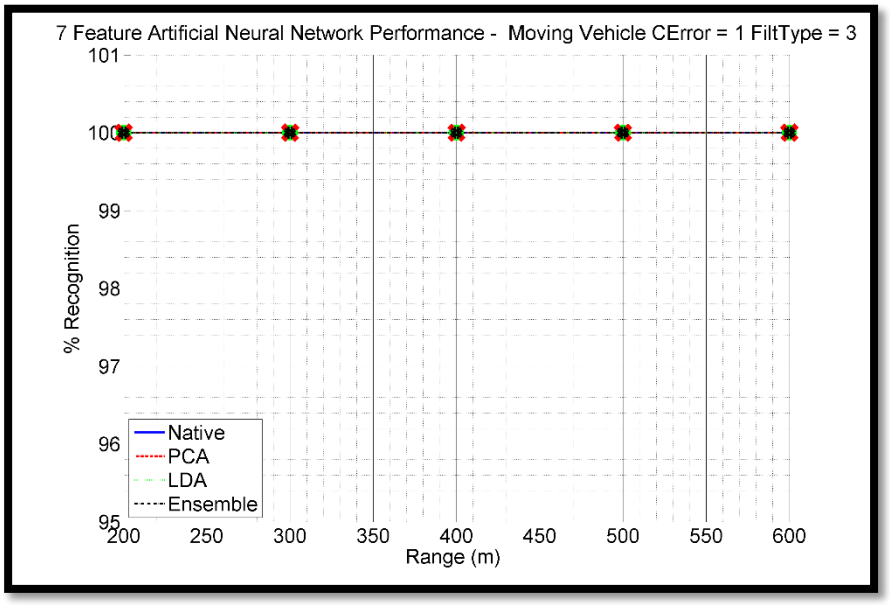

Figure 80 Percent Recognition of Vehicle Types -vs- Range with Seven Feature ANN: Moving Vehicle - Noise Level of 50 microradians - Filter Type 3

Noise Level of 100 microradians

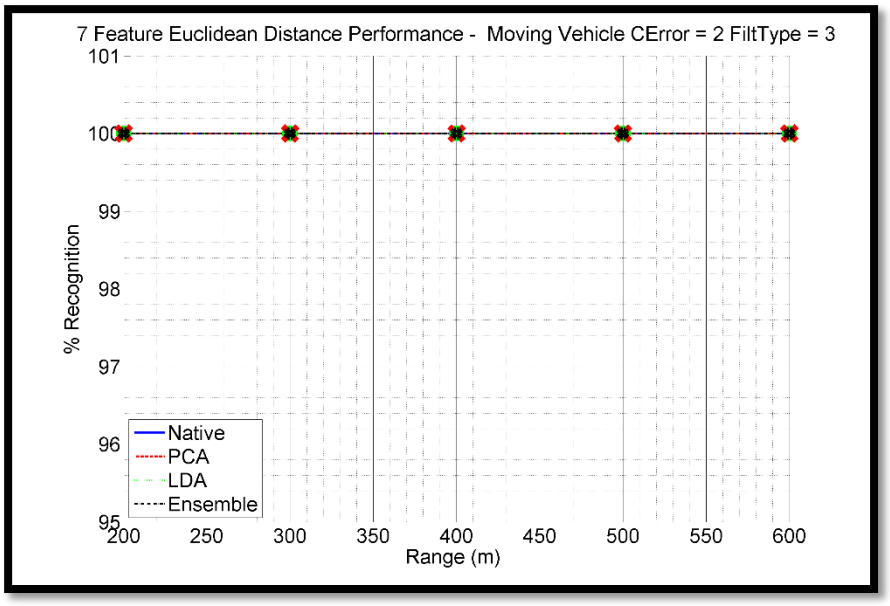

Figure 81 Percent Recognition of Vehicle Types -vs- Range with Seven Feature Euclidean Distance: Moving Vehicle - Noise Level of 100 microradians - Filter Type 3

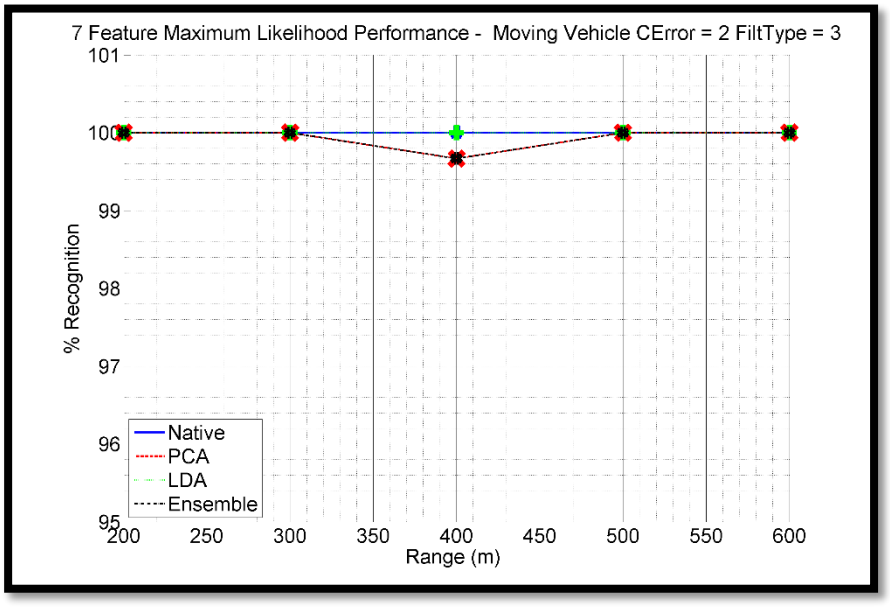

Figure 82 Percent Recognition of Vehicle Types -vs- Range with Seven Feature Max Likelihood: Moving Vehicle - Noise Level of 100 microradians – Filter Type 3

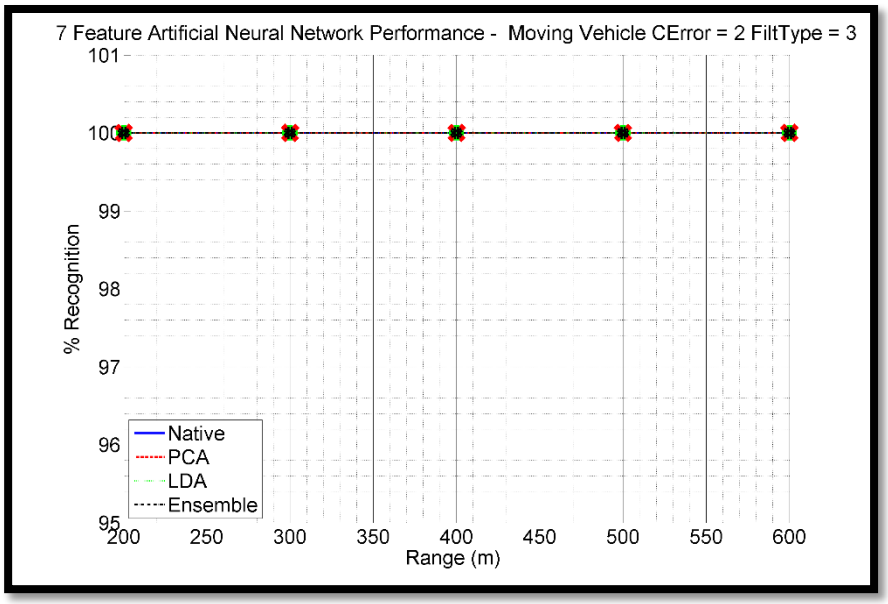

Figure 83 Percent Recognition of Vehicle Types -vs- Range with Seven Feature ANN: Moving Vehicle - Noise Level of 100 microradians - Filter Type 3

Noise Level of 150 microradians

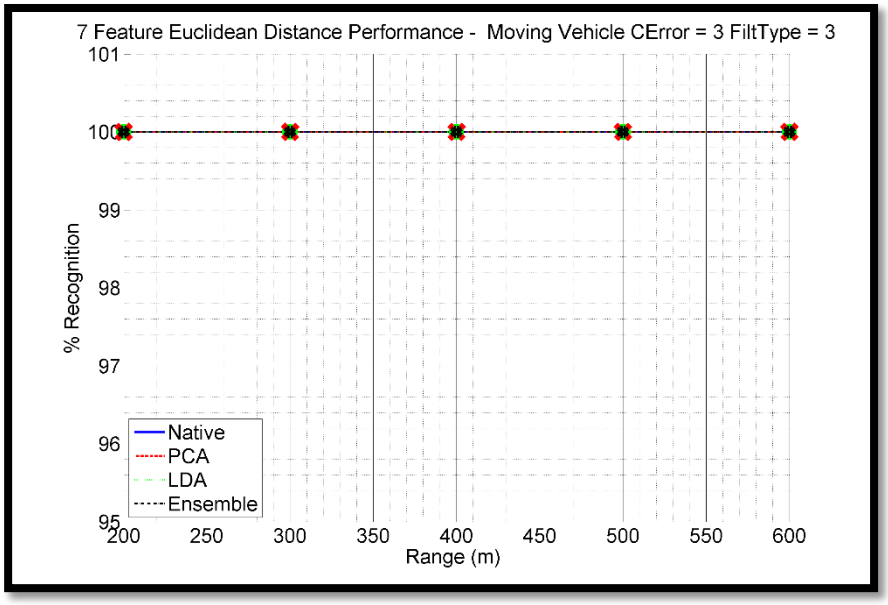

Figure 84 Percent Recognition of Vehicle Types -vs- Range with Seven Feature Euclidean Distance: Moving Vehicle - Noise Level of 150 microradians - Filter Type 3

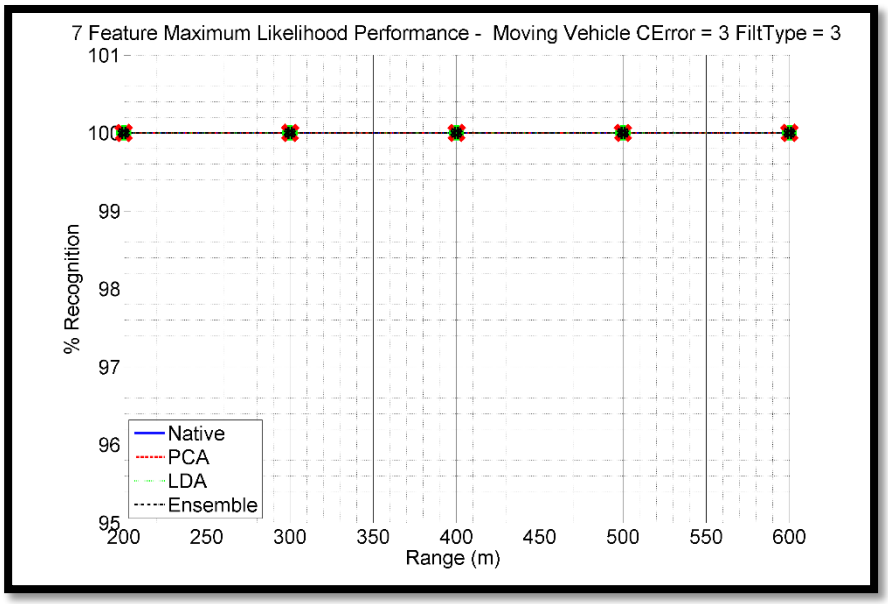

Figure 85 Percent Recognition of Vehicle Types -vs- Range with Seven Feature Max Likelihood: Moving Vehicle - Noise Level of 150 microradians – Filter Type 3

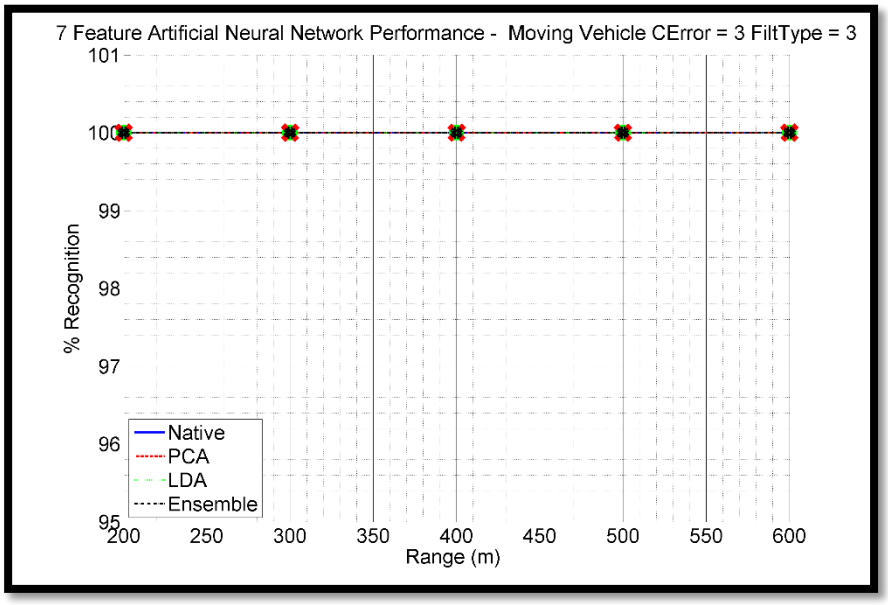

Figure 86 Percent Recognition of Vehicle Types -vs- Range with Seven Feature ANN: Moving Vehicle - Noise Level of 150 microradians - Filter Type 3

Noise Level of 200 microradians

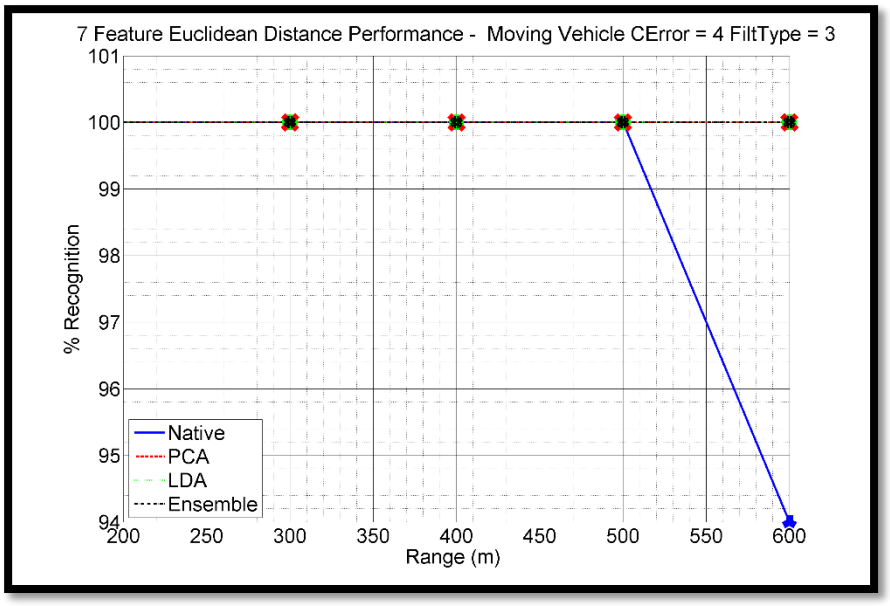

Figure 87 Percent Recognition of Vehicle Types -vs- Range with Seven Feature Euclidean Distance: Moving Vehicle - Noise Level of 200 microradians - Filter Type 3

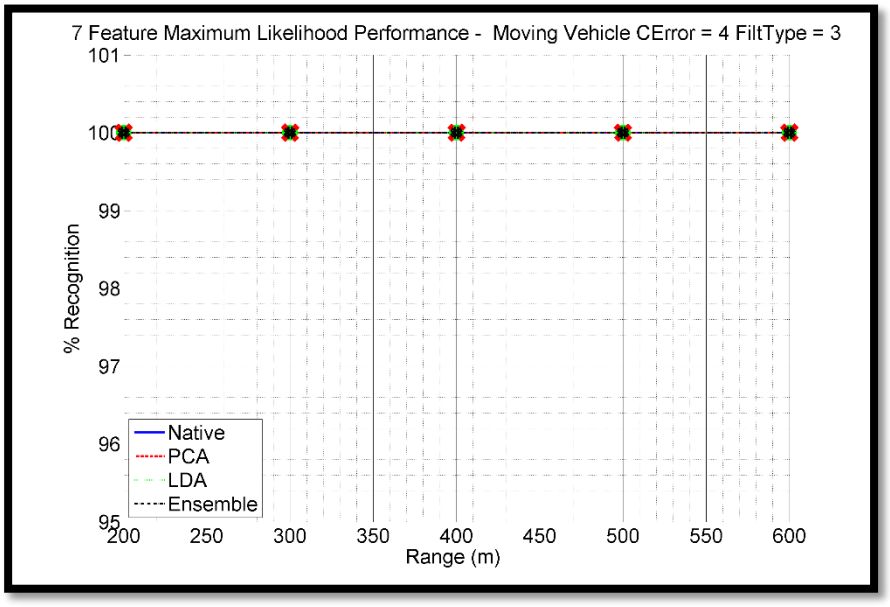

Figure 88 Percent Recognition of Vehicle Types -vs- Range with Seven Feature Max Likelihood: Moving Vehicle - Noise Level of 200 microradians – Filter Type 3

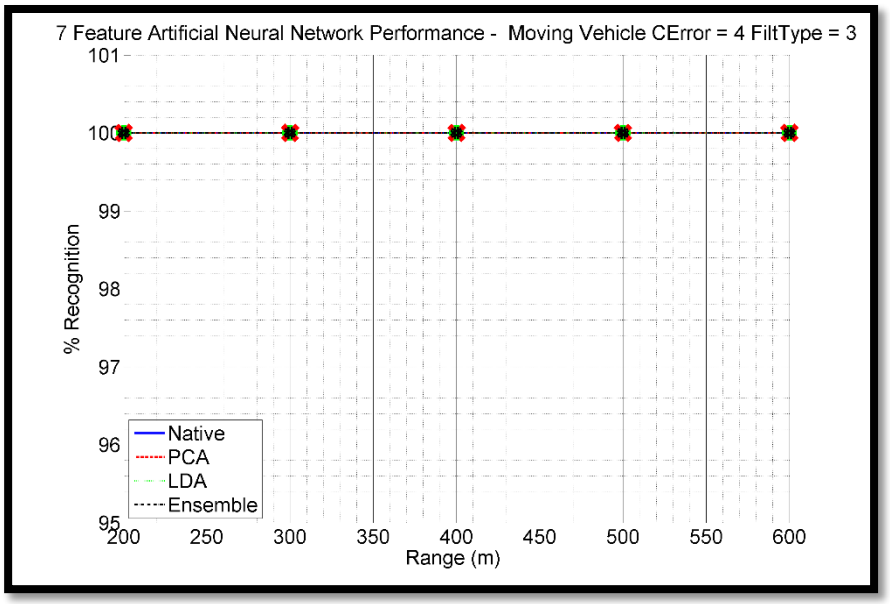

Figure 89 Percent Recognition of Vehicle Types -vs- Range with Seven Feature ANN: Moving Vehicle - Noise Level of 200 microradians - Filter Type 3

Seven Feature Conclusions

The performance for the seven feature classification performance averaged across all ranges and noise levels for both Filter Type 1 and Filter Type 3 is shown in Table 8 and Table 9.

Table 8
Seven Feature Correct Classification Percentage For Filter Type 1 Averaged Across All Range and Noise Levels

Approach	% Correct
PCA ANN	89.77
ANN Ensemble	89.05
Max Likelihood Ensemble	86.37
PCA Max Likelihood	85.72
LDA Max Likelihood	85.18
Native ANN	85.17
Euclidean Ensemble	84.38
LDA Euclidean	84.27
LDA ANN	80.87
Native Max Likelihood	79.47
PCA Euclidean	68.17
Native Euclidean	64.03

Table 9
Seven Feature Correct Classification Percentage For Filter Type 3 Averaged Across All Ranges and Noise Levels

Approach	% Correct
Native ANN	100.00
PCA Max Likelihood	100.00
PCA ANN	100.00
LDA Euclidean	100.00
LDA Max Likelihood	100.00
LDA ANN	100.00
Euclidean Ensemble	100.00
Max Likelihood Ensemble	100.00
ANN Ensemble	100.00
Native Max Likelihood	99.98
PCA Euclidean	99.98
Native Euclidean	99.70

Filter Type 1, the three six state filters, did well initially, but performance dropped off significantly as range increased. This indicates Filter Type 1 is sensitive to the effects of noise, which are linearly amplified as a function of distance. Intuitively, this is consistent with the model being used to construct the vectors in 3D space. Given that the 3D location of each PoI is being reconstructed independently, and the noise on each angle measurement for each point is independent, the predicted geometries can easily diverge in the presence of significant noise and lead to highly skewed feature values. Filter Type 3, the combined twelve state filter, consistently performed well, thus demonstrating its robustness for dealing with noise and added range. This was a result of the twelve state filters a priori knowledge of the rigid geometry connecting the PoI which served to constrain the geometry to its true form. This outcome was confirmed by both the plotted data and table of average results. Consequently, we conclude Filter Type 3 is the superior solution for estimating the required parameters for implementing the seven feature approach.

Filter Type 3 did not provide much insight into the effects of evaluating the features in different transform domains, simply because of how well all of the classifiers did in all of the data domains. Conversely, Filter Type 1 provided more varied results for the purpose of evaluating the utility of transforming the features with LDA or PCA prior to evaluation in the classifiers. Two of the three native spatial domain approaches scored in the bottom three in terms of classification performance. Additionally, the native spatial domain approach that did not score in the bottom three was the approach which implemented the feed forward neural network for classification. As noted earlier in the text, the neural network maps the native spatial domain data into a new domain before classification occurs. Essentially, the neural network is performing a domain transform operation similar to PCA or LDA prior to

classification. Therefore, we conclude the native spatial domain features can be more effectively represented in a transform domain for the purposes of classification.

Again, Filter Type 3 did not provide much insight into the effects of evaluating the different classification approaches due to the high performance levels across the board. Conversely, Filter Type 1 results were more varied and did provide insight into the performance of the different classification approaches. The top two using Filter Type 1 were ANN based classifiers implemented using transform domain features. A close third in terms of performance was the maximum likelihood ensemble approach. The maximum likelihood ensemble approach implemented a maximum likelihood classifier for both the LDA and PCA domain features and then fused the results of the independent classifiers into a single result using a simple Bayesian Network. Consequently, the ensemble approach benefited from the advantages of using both the LDA and PCA domain features. Ultimately, ANN transform domain based approach provided the best solution for seven feature classification; however if an alternative approach is required, the maximum likelihood ensemble classifier would be a suitable pick.

CHAPTER SEVEN:
SIXTEEN FEATURE CLASSIFICATION OF VEHICLES

Euler angles provide a mechanism to represent orthogonal angular information with separate values [128][129][130]. Additionally, the dot product formula provides a convenient mechanism for characterizing the orientation between two vectors with only a single value. Both approaches present valuable sources of information if handled correctly. However, the values of the Euler angles and dot product angles are dependent on the reference frame to which they are relative. For the purposes of classification, one must choose a reference frame in which the Euler angles and dot products are constant. For the sixteen feature approach detailed in this section, the reference frame to which the Euler angles and dot products are referenced will be calculated using the velocity vector of PoI A of the vehicle. The velocity reference frame is referred to as frame V. The X-Axis of the reference frame V will be aligned with the velocity vector of PoI A of the vehicle. They Y-Axis and Z-Axis will be calculated using cross product operations and when combined with the X-Axis will result in a right handed coordinate system to serve as reference frame V. The velocity vector of point A is assumed to have a constant orientation to the vectors between PoI A and PoI B, PoI A and PoI C and PoI B and PoI C on the vehicle regardless of vehicle orientation. It's worth re-noting, one of the requirements for the proposed techniques are the velocity vector of the vehicle is constant. In other words, the vehicle can't be turning or changing lanes. Given that the velocity vector is being used a reference frame, the sixteen feature approach is only applicable to moving vehicles. Equation 252 through Equation 267 demonstrates the calculation of the features for the sixteen feature classification approach.

$$\boldsymbol{\theta_{AB_i}^V = \sin^{-1}\left(\vec{U}_{AB_{z_i}}^V\right)} \quad (252)$$

$$\boldsymbol{\psi_{AB_i}^V = \tan^{-1}\left(\frac{\vec{X}_{AB_{y_i}}^V}{\vec{X}_{AB_{x_i}}^V}\right)} \quad (253)$$

$$\boldsymbol{r_{AB_i} = \sqrt{\left(\vec{X}_B^R - \vec{X}_A^R\right)^2}} \quad (254)$$

$$\boldsymbol{\theta_{AC_i}^V = \sin^{-1}\left(\vec{U}_{AB_{z_i}}^V\right)} \quad (255)$$

$$\boldsymbol{\psi_{AC_i}^V = \tan^{-1}\left(\frac{\vec{X}_{AC_{y_i}}^V}{\vec{X}_{AC_{x_i}}^V}\right)} \quad (256)$$

$$\boldsymbol{r_{AC_i} = \sqrt{\left(\vec{X}_C^R - \vec{X}_A^R\right)^2}} \quad (257)$$

$$\boldsymbol{\theta_{BC_i}^V = \sin^{-1}\left(\vec{U}_{BC_{z_i}}^V\right)} \quad (258)$$

$$\boldsymbol{\psi_{BC_i}^V = \tan^{-1}\left(\frac{\vec{X}_{BC_{y_i}}^V}{\vec{X}_{BC_{x_i}}^V}\right)} \quad (259)$$

$$\boldsymbol{r_{BC_i} = \sqrt{\left(\vec{X}_B^R - \vec{X}_C^R\right)^2}} \quad (260)$$

$$\boldsymbol{\theta_{AC_i}^{AB} = \sin^{-1}\left(\vec{U}_{AC_{z_i}}^{AB}\right)} \quad (261)$$

$$\boldsymbol{\psi_{AC_i}^{AB}} = \mathbf{tan^{-1}}\left(\frac{\vec{X}_{AC_{y_i}}^{AB}}{\vec{X}_{AC_{x_i}}^{AB}}\right) \quad (262)$$

$$\boldsymbol{\theta_{BC_i}^{AB}} = \mathbf{sin^{-1}}\left(\vec{U}_{BC_{z_i}}^{AB}\right) \quad (263)$$

$$\boldsymbol{\psi_{BC_i}^{AB}} = \mathbf{tan^{-1}}\left(\frac{\vec{X}_{BC_{y_i}}^{AB}}{\vec{X}_{BC_{x_i}}^{AB}}\right) \quad (264)$$

$$\boldsymbol{\gamma_{AB}} = \mathbf{cos^{-1}}\left(\frac{\vec{X}_{AB}^{R} \cdot \vec{V}_{A}^{R}}{|\vec{X}_{AB}^{R}| * |\vec{V}_{A}^{R}|}\right) \quad (265)$$

$$\boldsymbol{\gamma_{AC}} = \mathbf{cos^{-1}}\left(\frac{\vec{X}_{AC}^{R} \cdot \vec{V}_{A}^{R}}{|\vec{X}_{AC}^{R}| * |\vec{V}_{A}^{R}|}\right) \quad (266)$$

$$\boldsymbol{\gamma_{BC}} = \mathbf{cos^{-1}}\left(\frac{\vec{X}_{BC}^{R} \cdot \vec{V}_{A}^{R}}{|\vec{X}_{BC}^{R}| * |\vec{V}_{A}^{R}|}\right) \quad (267)$$

The sixteen feature classification approach was implemented using two separate Kalman Filter formulations. The first formulation, referred to as Filter Type 1, employed three separate six state filters which each independently tracked a PoI. Using the 3D position information contained in each six state filter for their respective PoI, the vectors and angles of interest were calculated. The second formulation, referred to as Filter Type 3, employed a single twelve state filter which use knowledge of the feature geometry to track all three PoI as a single rigid structure. Using the 3D position information for the PoI contained in the

twelve state filter, the vectors and angles of interest were calculated. The symbols corresponding to the different domain data types tested in each classifier are depicted in Figure 90.

Data Domain	Symbol
Native Spatial	
PCA	
LDA	
Ensemble	

Figure 90 Data Domain Sixteen Feature Classification Plot Symbols

Experimental Results – Filter Type 1 - Six State Filter

The classification performance of Filter Type 1 for the sixteen feature approach is detailed in this section for constant velocity vehicles with varying levels of noise.

Noise Level of 50 microradians

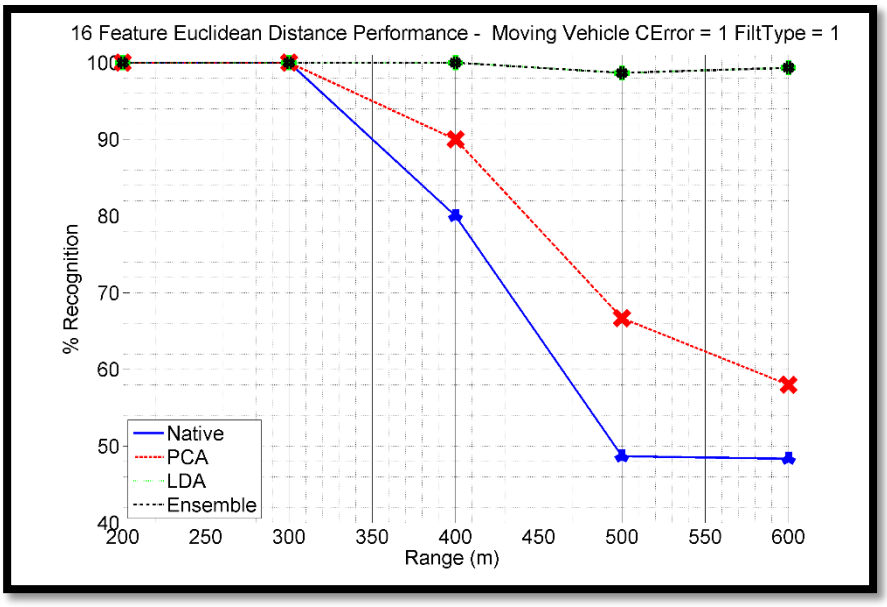

Figure 91 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature Euclidean Distance: Moving Vehicle - Noise Level of 50 microradians - Filter Type 1

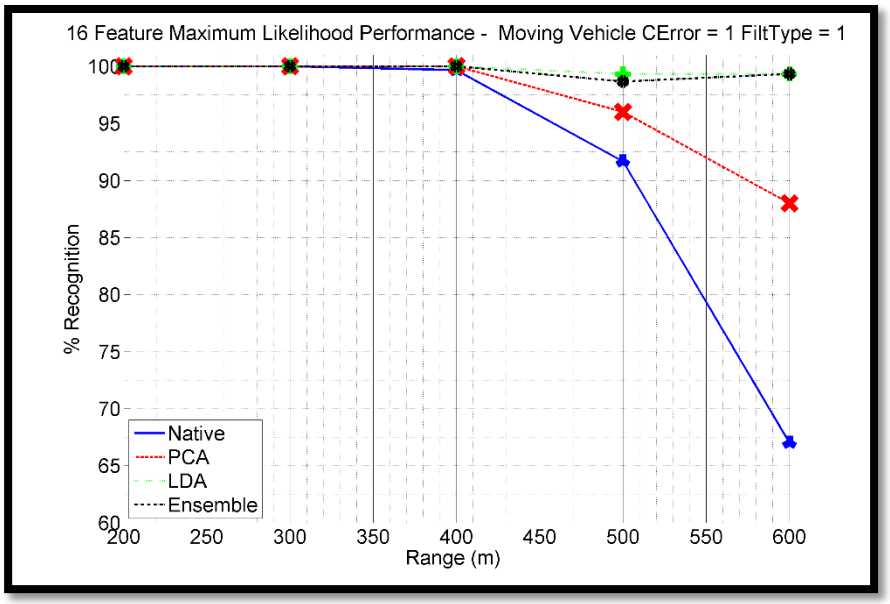

Figure 92 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature Max Likelihood: Moving Vehicle - Noise Level of 50 microradians – Filter Type 1

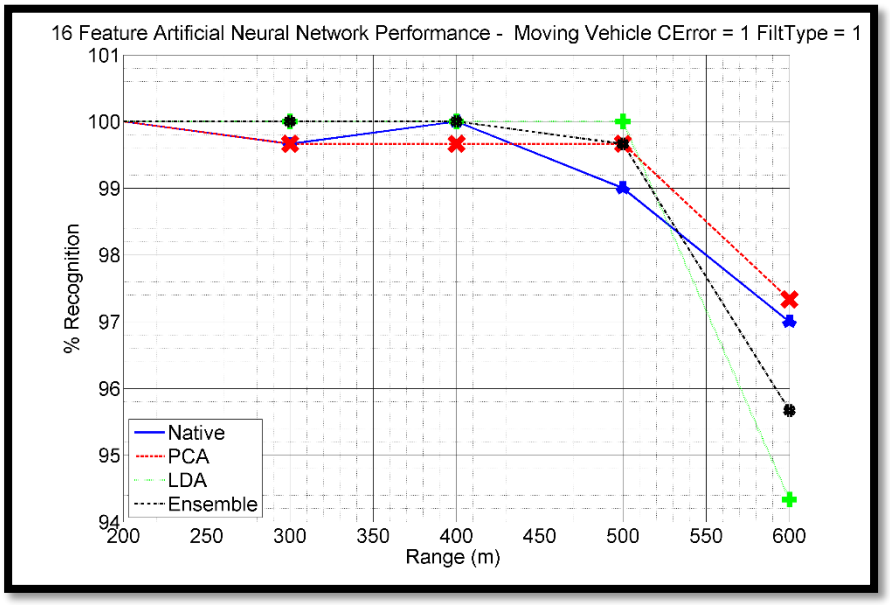

Figure 93 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature ANN: Moving Vehicle - Noise Level of 50 microradians - Filter Type 1

Noise Level of 100 microradians

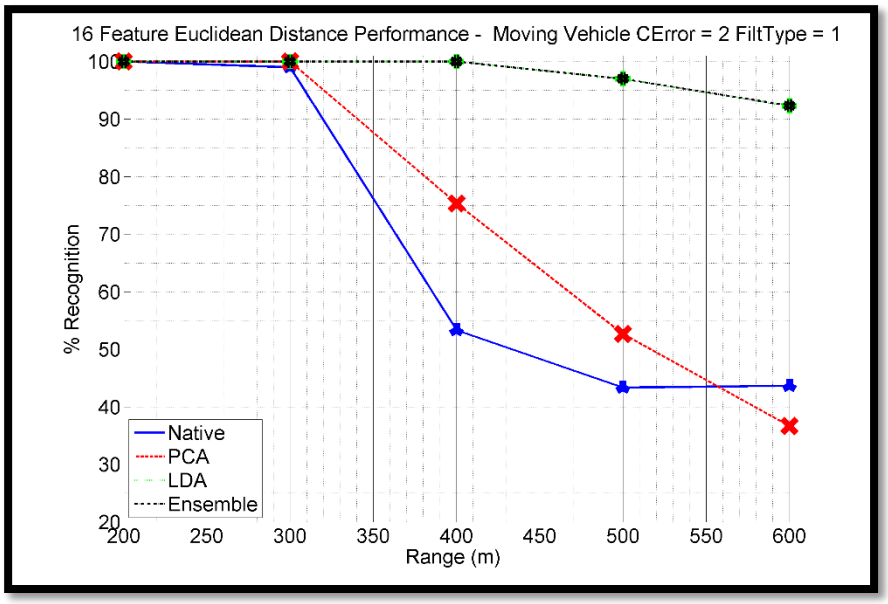

Figure 94 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature Euclidean Distance: Moving Vehicle - Noise Level of 100 microradians - Filter Type 1

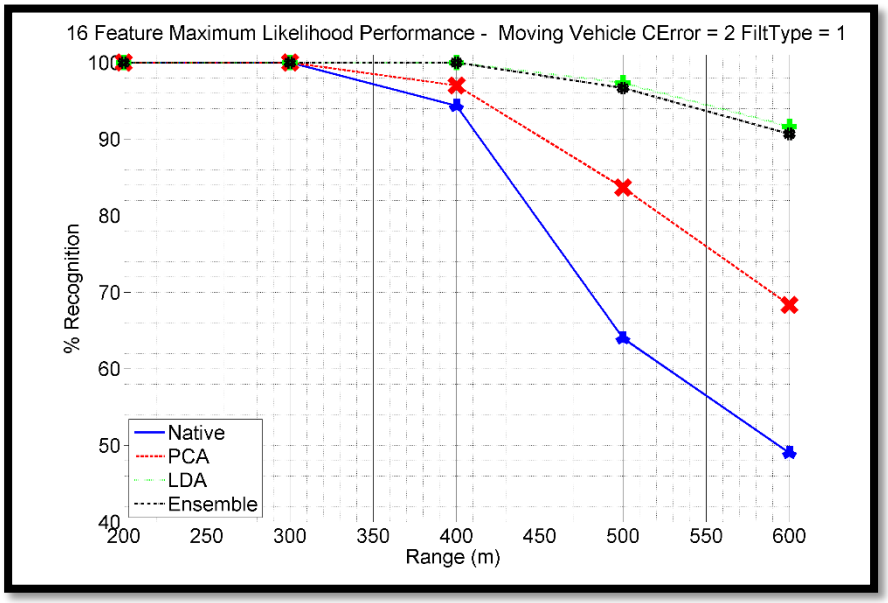

Figure 95 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature Max Likelihood: Moving Vehicle - Noise Level of 100 microradians – Filter Type 1

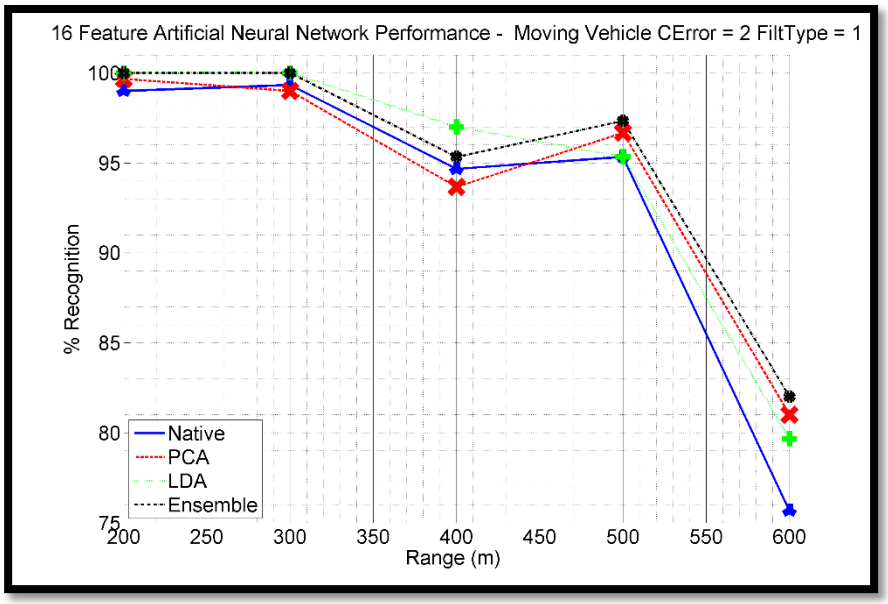

Figure 96 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature ANN: Moving Vehicle - Noise Level of 100 microradians - Filter Type 1

Noise Level of 150 microradians

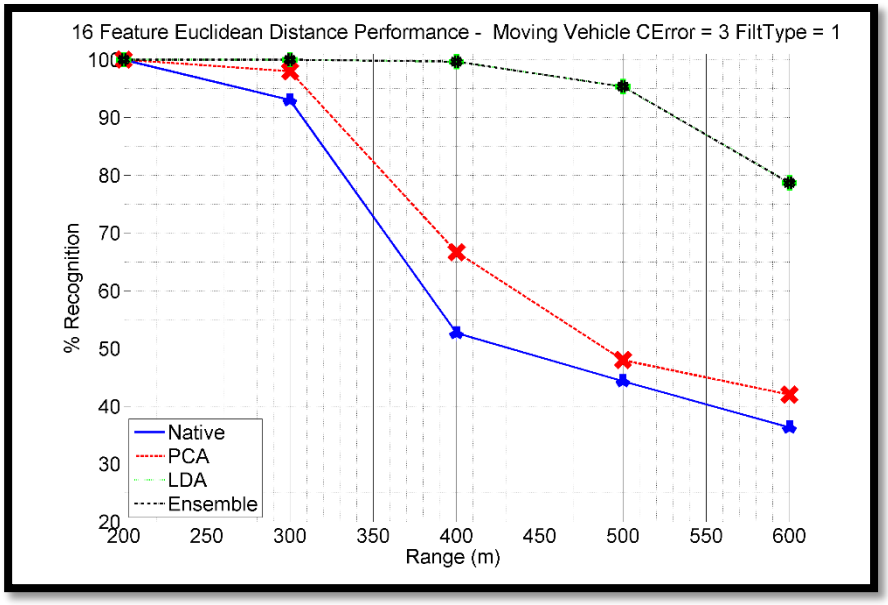

Figure 97 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature Euclidean Distance: Moving Vehicle - Noise Level of 150 microradians - Filter Type 1

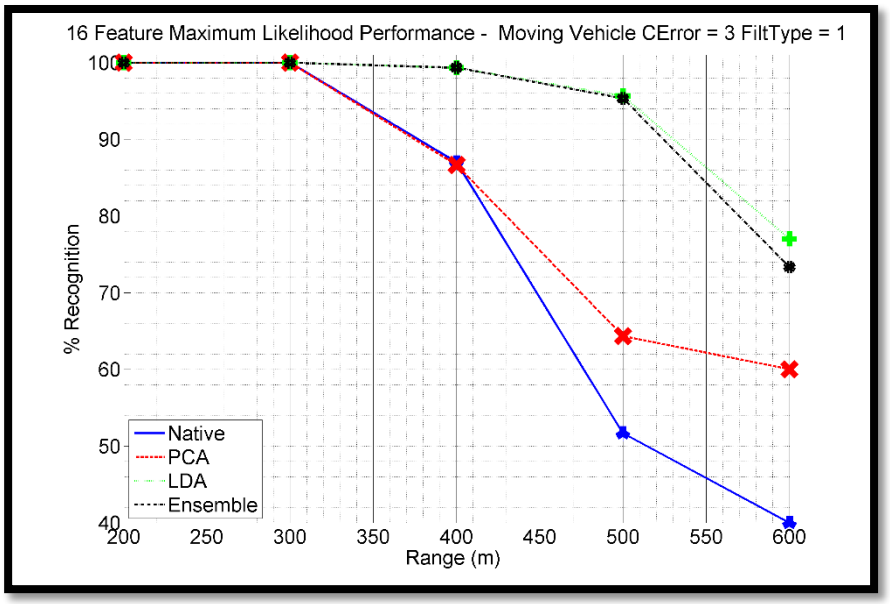

Figure 98 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature Max Likelihood: Moving Vehicle - Noise Level of 150 microradians – Filter Type 1

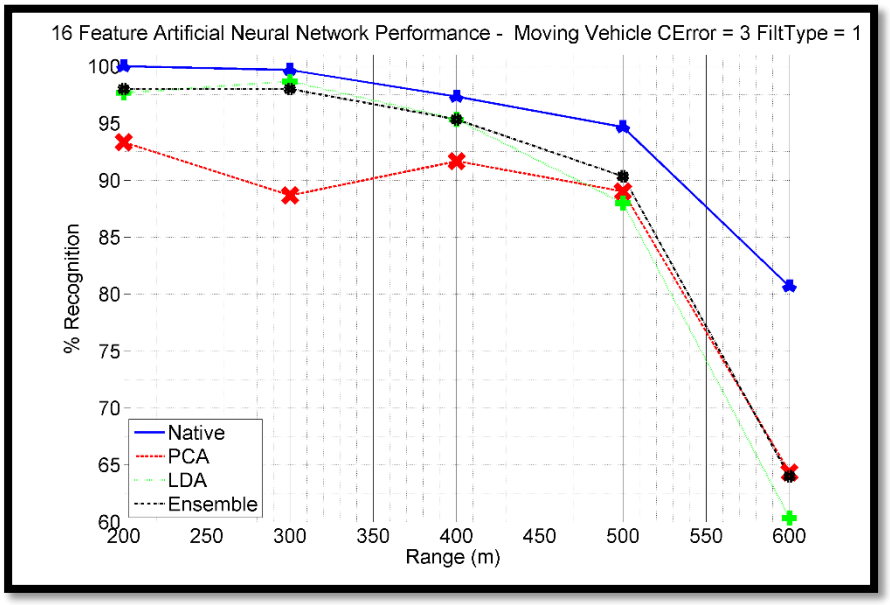

Figure 99 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature ANN: Moving Vehicle - Noise Level of 150 microradians - Filter Type 1

Noise Level of 200 microradians

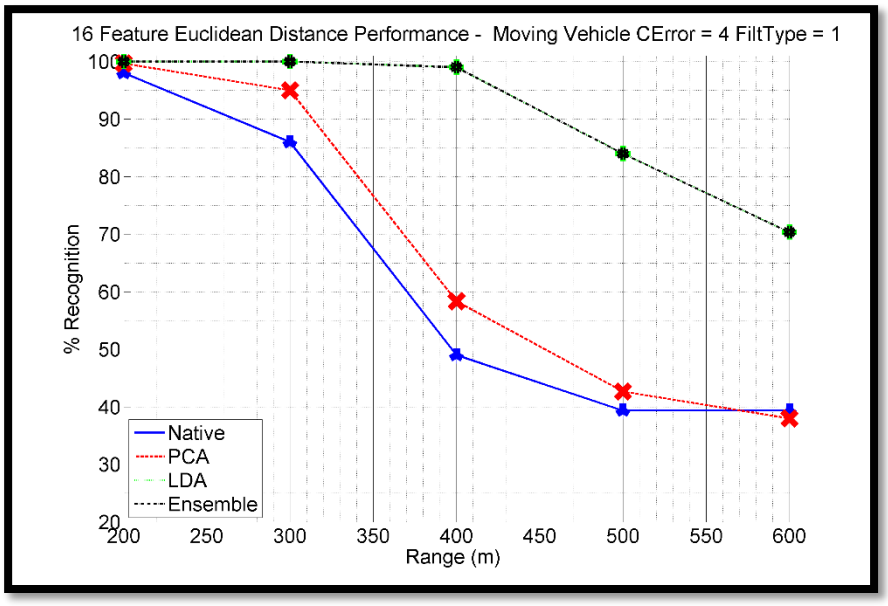

Figure 100 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature Euclidean Distance: Moving Vehicle - Noise Level of 200 microradians- Filter Type 1

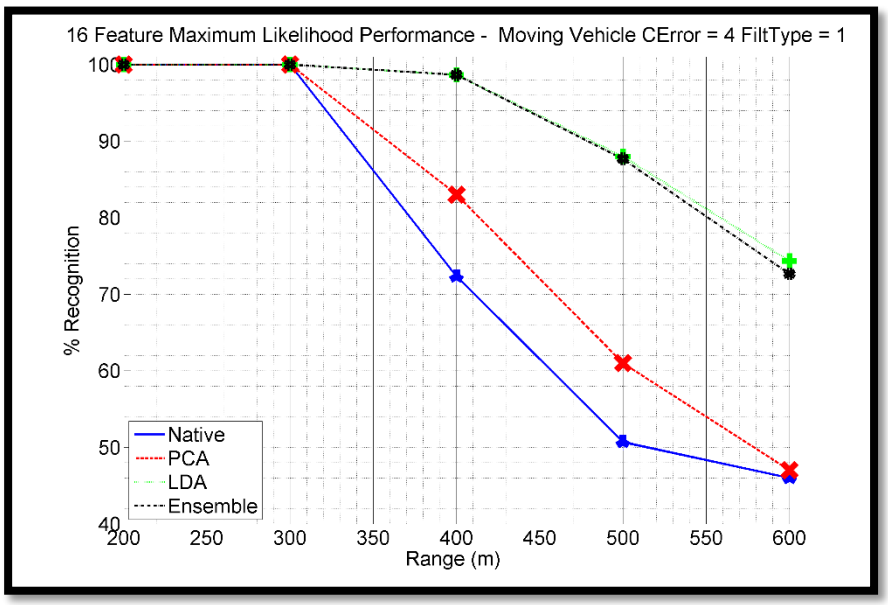

Figure 101 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature Max Likelihood: Moving Vehicle - Noise Level of 200 microradians – Filter Type 1

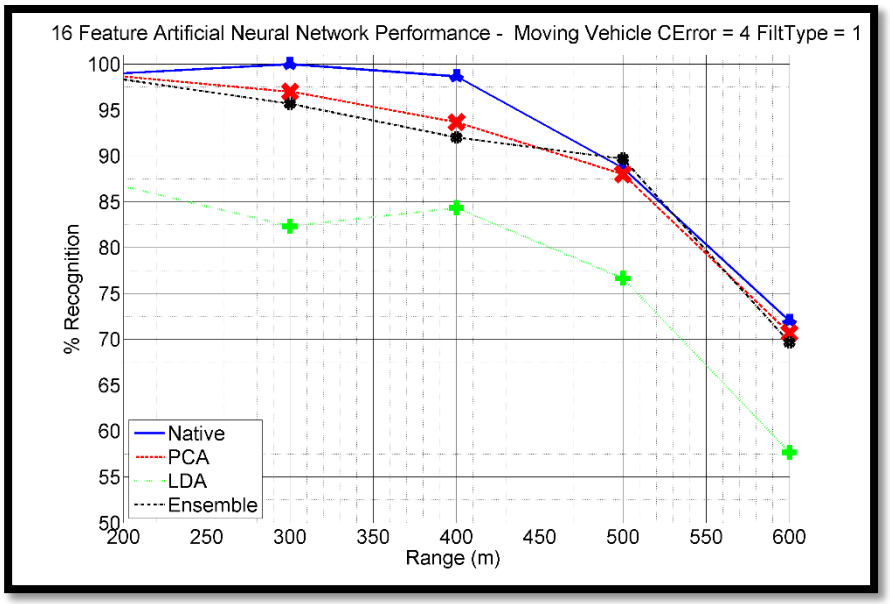

Figure 102 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature ANN: Moving Vehicle - Noise Level of 200 microradians - Filter Type 1

Experimental Results – Filter Type 3 - Twelve State Filter

The classification performance of Filter Type 3 for the sixteen feature approach is detailed in this section for constant velocity vehicles with varying levels of noise.

Noise Level of 50 microradians

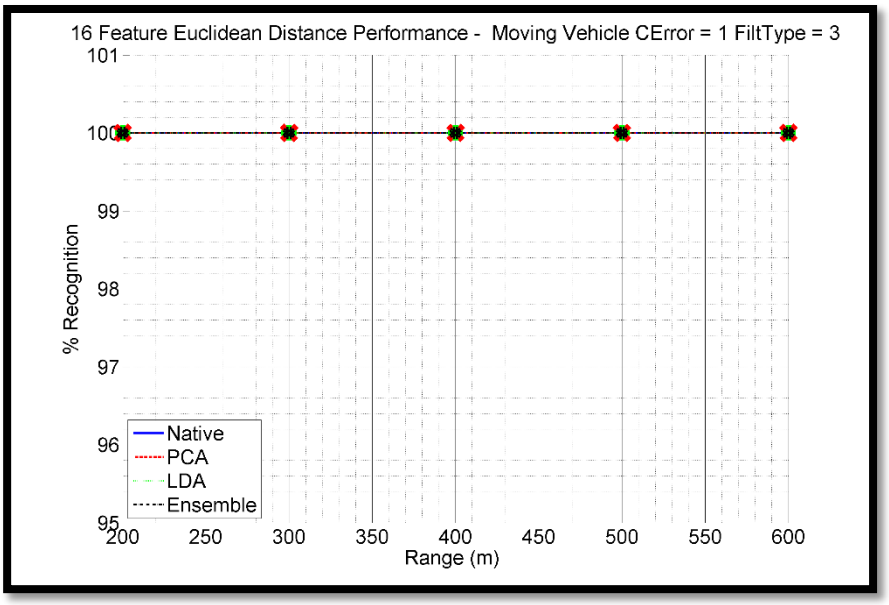

Figure 103 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature Euclidean Distance: Moving Vehicle - Noise Level of 50 microradians - Filter Type 3

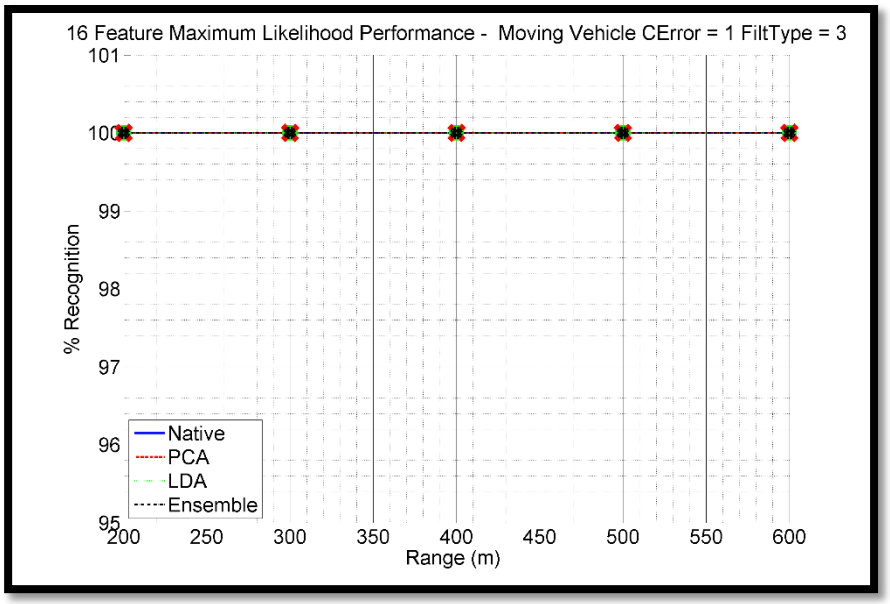

Figure 104 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature Max Likelihood: Moving Vehicle - Noise Level of 50 microradians – Filter Type 3

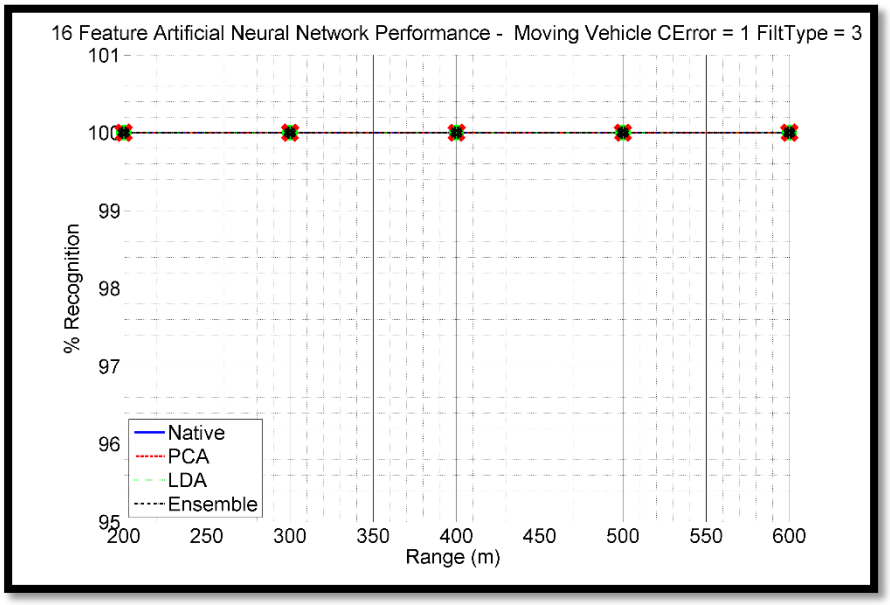

Figure 105 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature ANN: Moving Vehicle - Noise Level of 50 microradians - Filter Type 3

Noise Level of 100 microradians

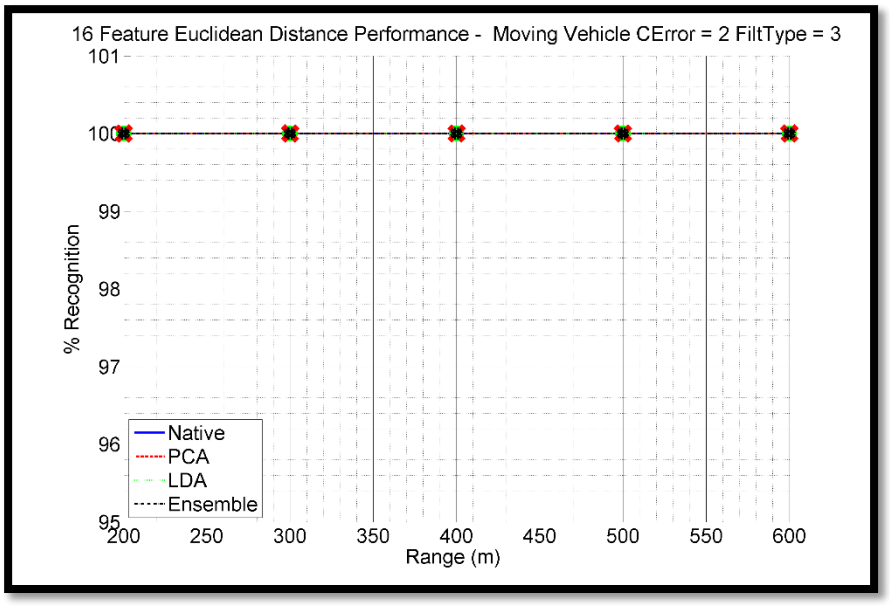

Figure 106 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature Euclidean Distance: Moving Vehicle - Noise Level of 100 microradians - Filter Type 3

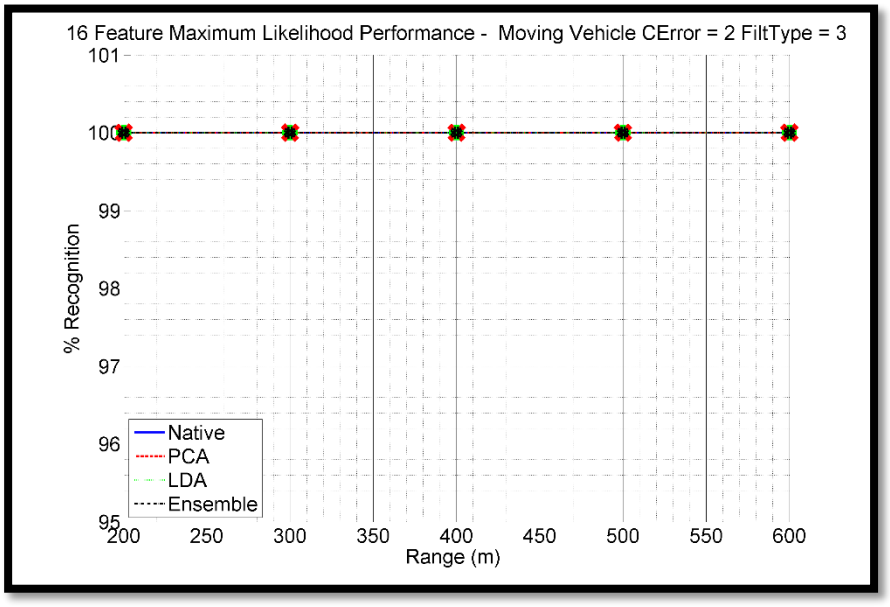

Figure 107 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature Max Likelihood: Moving Vehicle - Noise Level of 100 microradians – Filter Type 3

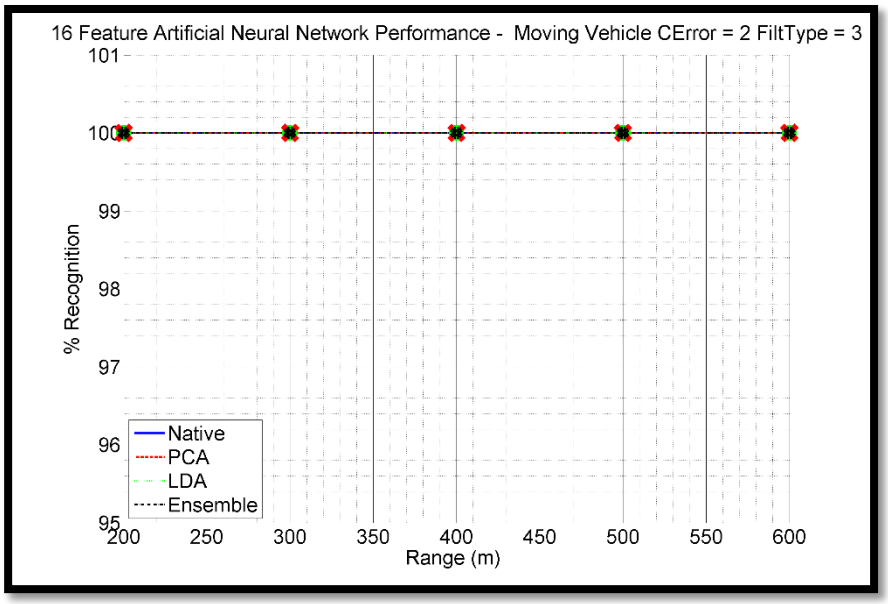

Figure 108 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature ANN: Moving Vehicle - Noise Level of 100 microradians - Filter Type 3

Noise Level of 150 microradians

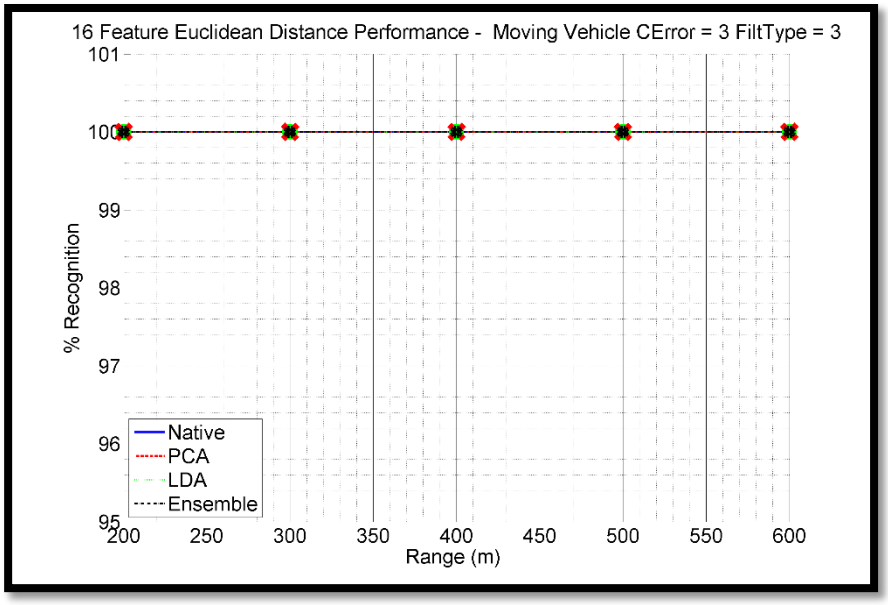

Figure 109 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature Euclidean Distance: Moving Vehicle - Noise Level of 150 microradians - Filter Type 3

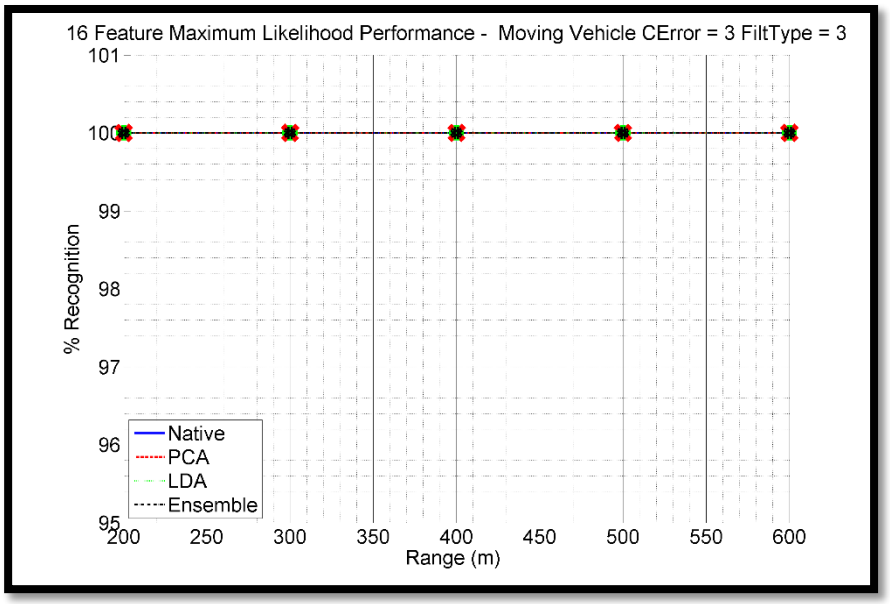

Figure 110 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature Max Likelihood: Moving Vehicle - Noise Level of 150 microradians – Filter Type 3

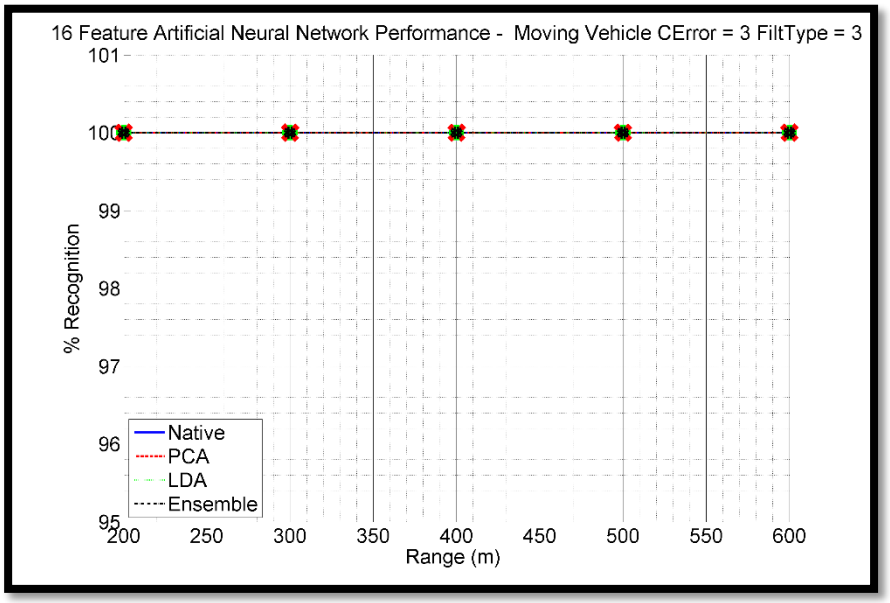

Figure 111 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature ANN: Moving Vehicle - Noise Level of 150 microradians - Filter Type 3

Noise Level of 200 microradians

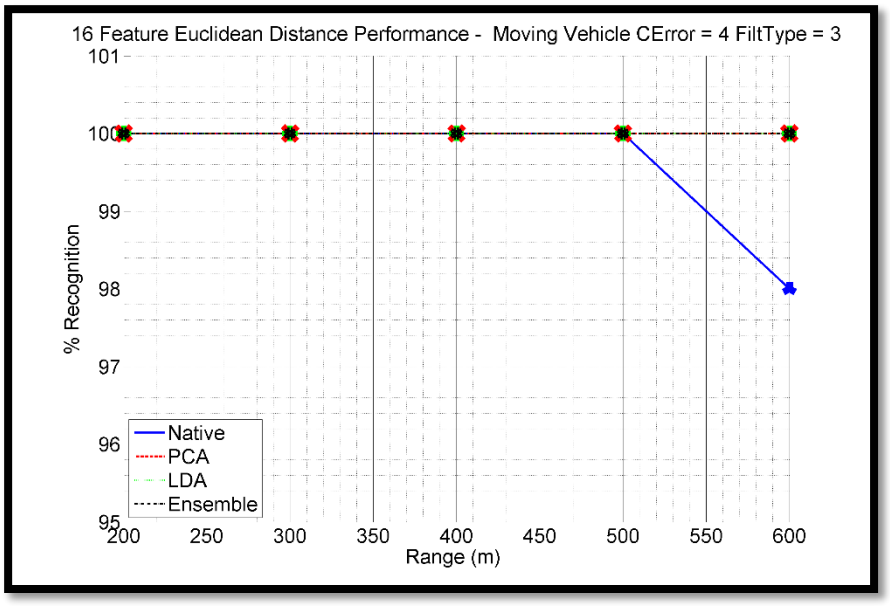

Figure 112 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature Euclidean Distance: Moving Vehicle - Noise Level of 200 microradians - Filter Type 3

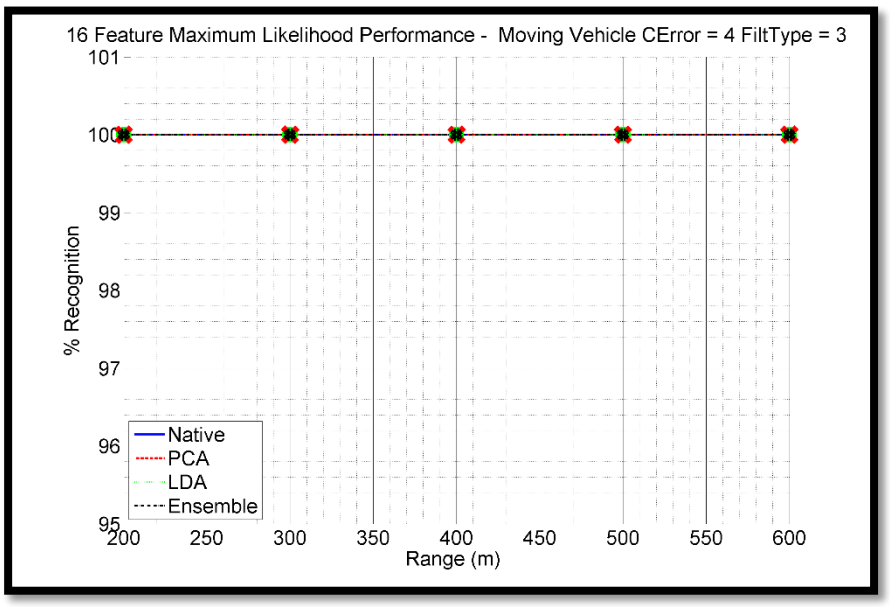

Figure 113 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature Max Likelihood: Moving Vehicle - Noise Level of 200 microradians – Filter Type 3

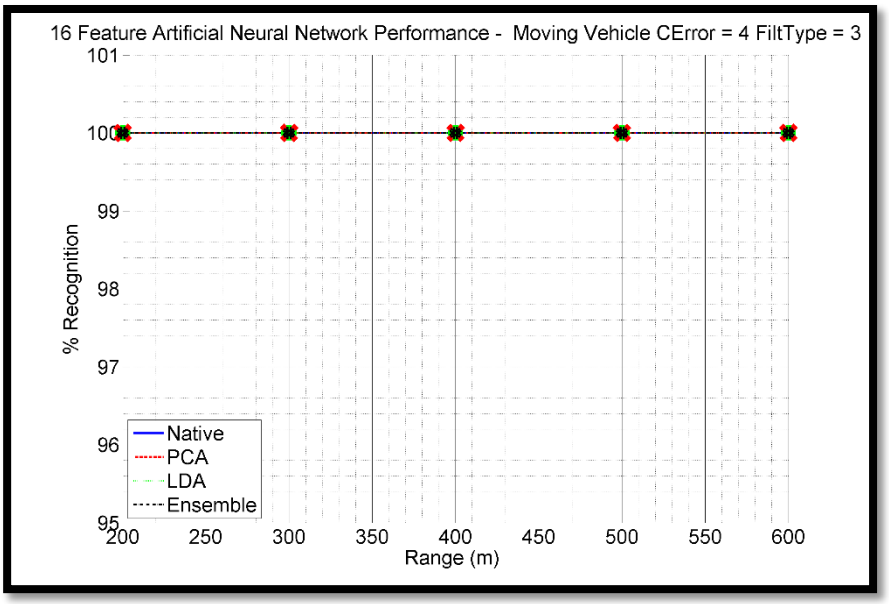

Figure 114 Percent Recognition of Vehicle Types -vs- Range with Sixteen Feature ANN: Moving Vehicle - Noise Level of 200 microradians - Filter Type 3

Sixteen Feature Conclusions

The performance for the sixteen feature classification performance averaged across all ranges and noise levels for Filter Type 1 and Filter Type 3 is shown in Table 10 and Table 11.

Table 10
Sixteen Feature Correct Classification Percentage For Filter Type 1 Averaged Across All Range and Noise Levels

Approach	% Correct
LDA Max Likelihood	96.03
LDA Euclidean	95.72
Euclidean Ensemble	95.72
Max Likelihood Ensemble	95.62
Native ANN	94.52
ANN Ensemble	93.05
PCA ANN	92.07
LDA ANN	89.70
PCA Max Likelihood	86.75
Native Max Likelihood	80.67
PCA Euclidean	73.38
Native Euclidean	67.72

Table 11
Sixteen Feature Correct Classification Percentage For Filter Type 2 Averaged Across All Ranges and Noise Levels

Approach	% Correct
Native Max Likelihood	100.00
Native ANN	100.00
PCA Euclidean	100.00
PCA Max Likelihood	100.00
PCA ANN	100.00
LDA Euclidean	100.00
LDA Max Likelihood	100.00
LDA ANN	100.00
Euclidean Ensemble	100.00
Max Likelihood Ensemble	100.00
ANN Ensemble	100.00
Native Euclidean	99.90

The robustness of Filter Type 1 varied as a function of the data domain in which the features were represented, as well as the classifier being used. The Euclidean Distance and Maximum Likelihood classifiers operating on features represented in the LDA domain demonstrated the highest performance and the least sensitivity to noise and increasing range. Additionally, the ensemble classifiers operating jointly on LDA and PCA data also demonstrated high performance and low sensitivity to noise and increasing range. Conversely, Minimum Euclidean Distance and Maximum Likelihood classifiers operating solely on features represented in either the native spatial or PCA domains degraded quickly in performance as range and noise levels increased.

Filter Type 3, the twelve state filter, performed extremely well, thus demonstrating its robustness for dealing with noise and increasing range. Its increased performance was a result of the twelve state filters a priori knowledge of the rigid geometry connecting the PoI, which served to constrain the geometry to its true form. Consequently, we conclude Filter Type 3 is the superior solution for estimating the required parameters to implement the sixteen feature approach.

Filter Type 3 results do not provide much insight into the effects of evaluating the features in different transform domains, simply because of how well all of the classifiers did in all of the data domains. Conversely, Filter Type 1 provides more varied results for the purpose of evaluating the utility of transforming the features with LDA or PCA prior to evaluation in the classifiers. Two of the three native spatial domain approaches scored in the bottom three in terms of classification performance, which indicates the LDA and PCA representations of the features are superior for classification purposes. Additionally, the remaining native spatial domain approach that did not score in the bottom three was the

approach which implemented the feed forward neural network for classification. As noted earlier in the text, the neural network maps the native spatial domain data into a new domain before classification occurs. Essentially, the neural network is performing a domain transform operation similar to PCA or LDA prior to classification. Consequently, we conclude the native spatial domain features can be more effectively represented in a transform domain resulting in superior classification performance.

Again, Filter Type 3 did not provide much insight into the effects of evaluating the different classification approaches due to the high performance levels across the board. Filter Type 1 results were more varied, but no clear conclusion can be reached as to which classification approach was superior. This is due to the fact the top five performers included all three classification approaches and there wasn't a significant difference in the level of performance between the five.

CHAPTER EIGHT:
CONCLUSSIONS AND FUTURE WORK

Conclusions

The choice of filter was the largest factor in determining the success of each of the proposed technique. Filter Type 2 and Filter Type 3 showed superior immunity to the deleterious effects of increased noise and range. Consequently, Filter Type 2 and Filter Type 3 where able to more consistently converge on the correct 3D representation of the PoI. This superior convergence capability translated into more accurate vector representations from which the classification features could be calculated. The classification approaches utilizing features represented in a transform domain consistently outperformed features represented in the native spatial domain. Therefore, we conclude the native spatial domain features can be more effectively represented in a transform domain for the purposes of classification. Maximum likelihood based techniques consistently demonstrated high performance relative to their peers across all four sets of features and also achieved the highest overall performance. Therefore, we conclude a maximum likelihood based approach to be the best choice based on both performance and complexity of implementation.

Overall, increasing the number of features did result in increasing classification performance. Table 12 lists the average classification percentage for all classification techniques for each set of proposed features averaged over all ranges and noise levels. The table shows a definite upward trend in average classification percentage as the number of features increases. Additionally, Table 13 lists the maximum classification percentage achieved for all classification techniques for each set of proposed features averaged over all

ranges and noise levels. Again, the table shows a definite upward trend in maximum classification percentage achieved as the number of features increases.

Inevitably, despite the outstanding performance of t Filter Type 2 and Filter Type 3, continuously increasing the noise and range will cause the performance to decline. The insights gained from Filter Type 1 results in regards to the use of transform domain features and varying classification approaches can provide guidance when determining the best transform domain / classifier combination to maximize the performance of Filter Type 2 and Filter Type 3.

Table 12
Average Classification Percentage For All Multipoint Techniques Averaged Across All Range and Noise Levels

Approach	Average % Correct For Filter Type 1	Average % Correct For Filter Type 2 or Filter Type 3
2 Feature	75.91	99.57
3 Feature	81.98	100.00
7 Feature	81.87	99.97
16 Feature	88.41	99.99

Table 13
Maximum Classification Percentage For All Multipoint Techniques Averaged Across All Range and Noise Levels

Approach	Max % Correct For Filter Type 1	Max % Correct For Filter Type 2 or Filter Type 3
2 Feature	81.05	100.00
3 Feature	86.87	100.00
7 Feature	89.77	100.00
16 Feature	96.03	100.00

One of the primary advantages of this approach, as opposed to purely image based approaches, is its ability to maintain a state vector which indicates the position and velocity of the vehicle even during data drop outs or missed detections. Such data drop outs or missed detections are inevitable under real world conditions. The state based approach will allow the methods proposed in this work to coast through those drop outs while still providing the vector angle and magnitude features needed for classification.

Future Work

Although the approaches presented in this work are specific to two or three PoI, the framework is easily extendable to as many PoI as desired, assuming the computational burden can be handled by the hardware. The framework can be altered to identify stationary targets simply by removing the velocity states from the filter derivation. Additionally, the approaches could also be altered to track and identify turning vehicles by incorporating acceleration states into the filter derivation. Implementing the techniques using particle filters as an alternative to Kalman filters to perform the optimal estimation presents an opportunity to better handle the non-linearities present in the tracking and measurement equations at the expense of greater computational overhead. Modifying the filter derivation to employ multiple cameras would enable a triangulation type approach for passively determining vehicle range. This change would allow for the proposed approaches to be implemented on fixed sensor installations such as roadside cameras. Additionally, fusing the outcome of these approaches with other classifiers which utilize orthogonal sensor modality information would increase the robustness and performance of the approaches.

Although the simulation used for this work had a great deal of fidelity, improvements can be made to account for additional real world factors. For example, the simulation did not model data drop outs or track miscorrelations. Also, additional sources of error can be modelled such as gimbal jitter or LOS reconstruction misalignments. Although the classification performance for some of the approaches achieved one hundred percent accuracy, it is highly unlikely this would occur in a real world situation. By adding additional fidelity to the simulation, the simulation performance will more closely mirror real world performance.

APPENDIX:
SYMBOL DEFINITION

Table 14
Symbol Definitions

Description	Symbol
Unit vector from boresight of Detector *D* to PoI *A* expressed in the *D* Frame.	$\vec{U}_{DA}^{D}$
Euler Rotation Matrix to rotate vector aligned with x-axis of *D* frame towards PoI *A*.	Φ_D
Euler Transformation matrix to transform vector from *D* frame to *R* frame.	C_D^R
Unit vector from boresight of Detector *D* to PoI *A* represented in *R* Frame.	$\vec{U}_{DA}^{R}$
Azimuth angle of unit vector from boresight of Detector *D* to PoI *A* represented in *R* Frame.	ψ_{DA}^{R}
Elevation angle of unit vector from boresight of Detector *D* to PoI *A* represented in *R* Frame.	θ_{DA}^{R}
6x1 vector representing State Estimate.	$\hat{X}$
3x1 vector representing position in State Estimate.	$\hat{X}_{DT}^{R}$
3x1 vector representing velocity in State Estimate.	$\hat{V}_{DT}^{R}$
State Transition Matrix at time k.	F_k
State Estimate at time k-1.	$\hat{X}_{k-1}$
State Estimate at time k given State Estimate at time k-1.	$\hat{X}_{k\|k-1}$
State Estimate at time k given measurement at time k.	$\hat{X}_{k\|k}$
State Covariance at time k – 1	P_{k-1}
State Covariance at time k given State Covariance at time k -1.	$P_{k\|k-1}$
State Covariance at time k given measurement at time k.	$P_{k\|k}$
Measurement residual at time k	y_k
Measurement vector at time k	z_k
Measurement mapping matrix at time k	H_k
Residual Covariance at time k	S_k
Kalman gain at time k	K_k
Vector between PoI *A* and PoI *B* expressed in the *R* frame.	$\vec{X}_{AB}^{R}$
Velocity vector of PoI *A* expressed in the *R* frame.	$\vec{V}_{A}^{R}$
Magnitude of vector $\vec{X}_{AB}^{R}$	r_{AB}
Dot product angle between $\vec{X}_{AB}^{R}$ and $\vec{V}_{A}^{R}$	γ_{AB}
Elevation Euler Angle of $\vec{X}_{AB}^{V}$ expressed in the *V* frame.	$\theta_{AB_i}^{V}$
Azimuth Euler Angle of $\vec{X}_{AB}^{V}$ expressed in the *V* frame.	$\psi_{AB_i}^{V}$
Perturbation or partial derivative	δ
Covariance Matrix	Σ

LIST OF REFERENCES

[1] L. E. Y. Mimbela and L. A. Klein, "Summary of vehicle detection and surveillance technologies used in intelligent transportation systems," 2000.

[2] P. McGowen and M. Sanderson, "Accuracy of pneumatic road tube counters," in *Proceedings of the 2011 Western District Annual Meeting, Anchorage, AK, USA*, 2011, pp. 10–13.

[3] L. Figueiredo, I. Jesus, J. T. Machado, J. Ferreira, and J. M. De Carvalho, "Towards the development of intelligent transportation systems," in *Intelligent transportation systems*, 2001, vol. 88, pp. 1206–1211.

[4] "Traffic Parameter Definitions." [Online]. Available: http://ntl.bts.gov/DOCS/96100/ch02/ch02_03.html. [Accessed: 04-Apr-2015].

[5] "Intrusive Technologies." [Online]. Available: http://www.civil.iitb.ac.in/tvm/1111_nptel/525_AutoLoop/plain/plain.html. [Accessed: 02-Apr-2015].

[6] J. Gajda, R. Sroka, M. Stencel, A. Wajda, and T. Zeglen, "A vehicle classification based on inductive loop detectors," in *Instrumentation and Measurement Technology Conference, 2001. IMTC 2001. Proceedings of the 18th IEEE*, 2001, vol. 1, pp. 460–464.

[7] S. Ritchie and C. Sun, *Individual Vehicle Speed Estimation Using Single Loop Inductive Waveforms*. Call No: UCB-ITS-PWP-99-14. California PATH Working Paper, California, 1999.

[8] S. Oh, S. G. Ritchie, and C. Oh, "Real time traffic measurement from single loop inductive signatures," in *81st Annual TRB Meeting, Washington, DC*, 2001.

[9] S. Meta and M. G. Cinsdikici, "Vehicle-classification algorithm based on component analysis for single-loop inductive detector," *Veh. Technol. IEEE Trans. On*, vol. 59, no. 6, pp. 2795–2805, 2010.

[10] D. D. Desai and S. B. Somani, "Evaluation of intrusive vehicle detection and classification technique for traffic management," in *India Conference (INDICON), 2014 Annual IEEE*, 2014, pp. 1–6.

[11] H. A. Oliveira, F. R. Barbosa, O. M. Almeida, and A. P. Braga, "A vehicle classification based on inductive loop detectors using artificial neural networks," in *Industry Applications (INDUSCON), 2010 9th IEEE/IAS International Conference on*, 2010, pp. 1–6.

[12] S. S. M. Ali, B. George, and L. Vanajakshi, "A magnetically coupled inductive loop sensing system for less-lane disciplined traffic," in *Instrumentation and Measurement Technology Conference (I2MTC), 2012 IEEE International*, 2012, pp. 827–832.

[13] S. S. M. Ali, B. George, and L. Vanajakshi, "Mutually coupled multiple inductive loop system suitable for heterogeneous traffic," *IET Intell. Transp. Syst.*, vol. 8, no. 5, pp. 470–478, 2013.

[14] M. J. Caruso and L. S. Withanawasam, "Vehicle detection and compass applications using AMR magnetic sensors," in *Sensors Expo Proceedings*, 1999, vol. 477.

[15] L. Yalong, L. Bin, and W. Zhiqiang, "A novel method of automatic vehicle detection based on active magnetic theory," in *2010 International Conference on Information, Networking and Automation (ICINA)*, 2010, vol. 2.

[16] J. Li, X. Xu, H. Zhao, Y. Hu, and T. Z. Qiu, "An energy-efficient sub-Nyquist sampling method based on compressed sensing in wireless sensor network for vehicle detection," in *Connected Vehicles and Expo (ICCVE), 2013 International Conference on*, 2013, pp. 314–319.
[17] S. Y. Cheung, S. Coleri, B. Dundar, S. Ganesh, C.-W. Tan, and P. Varaiya, "Traffic measurement and vehicle classification with single magnetic sensor," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1917, no. 1, pp. 173–181, 2005.
[18] A. Haoui, R. Kavaler, and P. Varaiya, "Wireless magnetic sensors for traffic surveillance," *Transp. Res. Part C Emerg. Technol.*, vol. 16, no. 3, pp. 294–306, 2008.
[19] L. Zhang, C. Haas, and S. L. Tighe, "Evaluating Weigh-In-Motion Sensing Technology for Traffic Data Collection," in *Annual Conference of the Transportation Association of Canada*, 2007, pp. 1–17.
[20] P. Barsanescu, P. Carlescu, and D. MIHAI STEFANESCU, "A new weigh-in-motion and traffic monitoring system," in *20th Conference on Measurement of Force, Mass and Torque (together with 3rd Conference on Pressure Measurement and 1st Conference on Vibration Measurement), International Measurement Confederation (IMEKO)*, 2007.
[21] J. Wang and M. Wu, "An overview of research on weigh-in-motion system," in *Fifth World Congress on Intelligent Control and Automation, 2004. WCICA 2004*, 2004, vol. 6, pp. 5241–5244 Vol.6.
[22] X. Chen, L. Guo, J. Yu, J. Li, and R. Liu, "Evaluating Innovative Sensors and Techniques for Measuring Traffic Loads," in *IEEE International Conference on Networking, Sensing and Control, 2008. ICNSC 2008*, 2008, pp. 1074–1079.
[23] N. Chen, Q. Li, F. Li, and Z. Jia, "A data processing algorithm based on vehicle weigh-in-motion systems," in *2013 Ninth International Conference on Natural Computation (ICNC)*, 2013, pp. 227–231.
[24] R. Zhang, W. Lv, and Y. Guo, "A Vehicle Weigh-in-Motion System Based on Hopfield Neural Network Adaptive Filter," in *2010 International Conference on Communications and Mobile Computing (CMC)*, 2010, vol. 3, pp. 123–127.
[25] M. Mangeas, S. Glaser, and V. Dolcemascolo, "Neural networks estimation of truck static weights by fusing weight-in-motion data," in *Proceedings of the Fifth International Conference on Information Fusion, 2002*, 2002, vol. 1, pp. 456–462 vol.1.
[26] W. Xue, D. Wang, and L. Wang, "Monitoring the Speed, Configurations, and Weight of Vehicles Using an In Situ Wireless Sensing Network," *IEEE Trans. Intell. Transp. Syst.*, vol. PP, no. 99, pp. 1–9, 2014.
[27] J. Xiao and P. Lv, "Application of Wavelet Transform in Weigh-in-Motion," in *International Workshop on Intelligent Systems and Applications, 2009. ISA 2009*, 2009, pp. 1–4.
[28] Y. Xuan, H. Meng, X. Wang, and H. Zhang, "A high-range-resolution microwave radar system for traffic flow rate measurement," in *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, 2005, pp. 880–885.
[29] J. Fang, H. Meng, H. Zhang, and X. Wang, "A low-cost vehicle detection and classification system based on unmodulated continuous-wave radar," in *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, 2007, pp. 715–720.
[30] I. Urazghildiiev, R. Ragnarsson, P. Ridderstrom, A. Rydberg, E. Ojefors, K. Wallin, P. Enochsson, M. Ericson, and G. Lofqvist, "Vehicle classification based on the radar

measurement of height profiles," *Intell. Transp. Syst. IEEE Trans. On*, vol. 8, no. 2, pp. 245–253, 2007.
[31] P.-J. Wang, C.-M. Li, C.-Y. Wu, and H.-J. Li, "A channel awareness vehicle detector," *Intell. Transp. Syst. IEEE Trans. On*, vol. 11, no. 2, pp. 339–347, 2010.
[32] S. J. Park, T. Y. Kim, S. M. Kang, and K. H. Koo, "A novel signal processing technique for vehicle detection radar," in *IEEE MTT-S International Microwave Symposium Digest*, 2003, vol. 1, pp. 607–610.
[33] M. Cherniakov, R. S. A. Raja Abdullah, P. Jancovic, and M. Salous, "Forward scattering micro sensor for vehicle classification," in *Radar Conference, 2005 IEEE International*, 2005, pp. 184–189.
[34] R. S. A. Abdullah, M. I. Saripan, and M. Cherniakov, "Neural network based for automatic vehicle classification in forward scattering radar," 2007.
[35] E. K. Walton, I. P. Theron, S. Gunawan, and L. Cai, "Moving vehicle range profiles measured using a noise radar," in *Antennas and Propagation Society International Symposium, 1997. IEEE., 1997 Digest*, 1997, vol. 4, pp. 2597–2600.
[36] Y. Jo, J. Choi, and I. Jung, "Traffic Information Acquisition System with Ultrasonic Sensors in Wireless Sensor Networks," *Int. J. Distrib. Sens. Netw.*, vol. 2014, 2014.
[37] S. Jeon, E. Kwon, and I. Jung, "Traffic measurement on multiple drive lanes with wireless ultrasonic sensors," *Sensors*, vol. 14, no. 12, pp. 22891–22906, 2014.
[38] H. Kim, J.-H. Lee, S.-W. Kim, J.-I. Ko, and D. Cho, "Ultrasonic vehicle detector for side-fire implementation and extensive results including harsh conditions," *Intell. Transp. Syst. IEEE Trans. On*, vol. 2, no. 3, pp. 127–134, 2001.
[39] B. Barbagli, I. Magrini, G. Manes, A. Manes, G. Langer, and M. Bacchi, "A distributed sensor network for real-time acoustic traffic monitoring and early queue detection," in *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on*, 2010, pp. 173–178.
[40] B. Barbagli, L. Bencini, I. Magrini, G. Manes, and A. Manes, "A real-time traffic monitoring based on wireless sensor network technologies," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*, 2011, pp. 820–825.
[41] J. Kato, "An attempt to acquire traffic density by using road traffic sound," in *Active Media Technology, 2005.(AMT 2005). Proceedings of the 2005 International Conference on*, 2005, pp. 353–358.
[42] S. Chen, Z. P. Sun, and B. Bridge, "Automatic traffic monitoring by intelligent sound detection," in *Intelligent Transportation System, 1997. ITSC'97., IEEE Conference on*, 1997, pp. 171–176.
[43] J. F. Forren and D. Jaarsma, "Traffic monitoring by tire noise," in *Intelligent Transportation System, 1997. ITSC'97., IEEE Conference on*, 1997, pp. 177–182.
[44] V. Tyagi, S. Kalyanaraman, and R. Krishnapuram, "Vehicular traffic density state estimation based on cumulative road acoustics," *Intell. Transp. Syst. IEEE Trans. On*, vol. 13, no. 3, pp. 1156–1166, 2012.
[45] J. Ploetner and M. M. Trivedi, "A multimodal framework for vehicle and traffic flow analysis," in *IEEE Intelligent Transportation Systems Conference, 2006. ITSC '06*, 2006, pp. 1507–1512.
[46] B. Lakshminarayanan and H. Qi, "Civilian target detection using hierarchical fusion," in *Applied Imagery and Pattern Recognition Workshop, 2005. Proceedings. 34th*, 2005, p. 6 pp.–178.

[47] A. Roy, N. Gale, and L. Hong, "Fusion of Doppler Radar and video information for automated traffic surveillance," in *12th International Conference on Information Fusion, 2009. FUSION '09*, 2009, pp. 1989–1996.
[48] G. Arr, C. Sun, and R. P. Ramachandran, "Fusion of wavelet transform and color information features for automatic vehicle reidentification in intelligent transportation systems," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04)*, 2004, vol. 5, pp. V–285–8 vol.5.
[49] Z.-M. Wang, Z.-X. Yang, Y. Chen, J.-P. Wu, and X. Ding, "Image fusion algorithm in Intelligent Transport System," in *2008 International Conference on Machine Learning and Cybernetics*, 2008, vol. 1, pp. 74–77.
[50] W. McDowell and W. B. Mikhael, "MASINT fusion of multispectral, hyperspectral amp; kinematic phenomenology," in *2014 IEEE 57th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2014, pp. 157–160.
[51] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1459–1472, Nov. 2004.
[52] T. Dharamadhat, K. Thanasoontornlerk, and P. Kanongchaiyos, "Tracking object in video pictures based on background subtraction and image matching," in *IEEE International Conference on Robotics and Biomimetics, 2008. ROBIO 2008*, 2009, pp. 1255–1260.
[53] L. Liu, G. Duan, H. Ai, and S. Lao, "An evaluation of boosted features for vehicle detection," in *2012 IEEE Intelligent Vehicles Symposium (IV)*, 2012, pp. 956–961.
[54] Y. B. Brahme and P. S. Kulkarni, "An Implementation of Moving Object Detection, Tracking and Counting Objects for Traffic Surveillance System," in *2011 International Conference on Computational Intelligence and Communication Networks (CICN)*, 2011, pp. 143–148.
[55] X. Pan, Y. Guo, and A. Men, "Traffic Surveillance System for Vehicle Flow Detection," in *Second International Conference on Computer Modeling and Simulation, 2010. ICCMS '10*, 2010, vol. 1, pp. 314–318.
[56] W.-K. For, K. Leman, H.-L. Eng, B.-F. Chew, and K.-W. Wan, "A multi-camera collaboration framework for real-time vehicle detection and license plate recognition on highways," in *Intelligent Vehicles Symposium, 2008 IEEE*, 2008, pp. 192–197.
[57] J. M. Manendez, L. Salgado, E. Rendon, and N. Garcia, "Motorway surveillance through stereo computer vision," in *IEEE 33rd Annual 1999 International Carnahan Conference on Security Technology, 1999. Proceedings*, 1999, pp. 197–202.
[58] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: real-time tracking of the human body," in *, Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, 1996*, 1996, pp. 51–56.
[59] F. Tao, L. Lin-sheng, and T. Qi-chuan, "A novel adaptive motion detection based on k-means clustering," in *2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT)*, 2010, vol. 3, pp. 136–140.
[60] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1778–1792, Nov. 2005.
[61] T. Bouwmans, F. El Baf, and B. Vachon, "Background modeling using mixture of gaussians for foreground detection-a survey," *Recent Pat. Comput. Sci.*, vol. 1, no. 3, pp. 219–237, 2008.

[62] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, 1999, vol. 2, p. -252 Vol. 2.
[63] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Background modeling and subtraction by codebook construction," in *Image Processing, 2004. ICIP'04. 2004 International Conference on*, 2004, vol. 5, pp. 3061–3064.
[64] Y. Iwasaki, "A method of robust moving vehicle detection for bad weather using an infrared thermography camera," in *Proceedings of the 2008 International Conference on Wavelet Analysis and Pattern Recognition, Hong Kong*, 2008, pp. 30–31.
[65] Y. Zou, H. Zhao, H. Shi, and Y. Wang, "A moving vehicle segmentation method based on clustering of feature points for tracking at urban intersection," in *Circuits and Systems (APCCAS), 2010 IEEE Asia Pacific Conference on*, 2010, pp. 120–123.
[66] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, "Vehicle Detection in Satellite Images by Parallel Deep Convolutional Neural Networks," in *2013 2nd IAPR Asian Conference on Pattern Recognition (ACPR)*, 2013, pp. 181–185.
[67] Y. Yang, F. Liu, P. Wang, P. Luo, and X. Liu, "Vehicle detection methods from an unmanned aerial vehicle platform," in *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, 2012, pp. 411–415.
[68] G. Mo and S. Zhang, "Vehicles detection in Traffic Flow," in *2010 Sixth International Conference on Natural Computation (ICNC)*, 2010, vol. 2, pp. 751–754.
[69] F. Oberti, S. Calcagno, M. Zara, and C. S. Regazzoni, "Robust tracking of humans and vehicles in cluttered scenes with occlusions," in *2002 International Conference on Image Processing. 2002. Proceedings*, 2002, vol. 3, pp. 629–632 vol.3.
[70] A. Broggi, P. Cerri, and P. C. Antonello, "Multi-resolution vehicle detection using artificial vision," in *2004 IEEE Intelligent Vehicles Symposium*, 2004, pp. 310–314.
[71] H. Huijie, X. Chao, Z. Jun, and G. Wenjun, "The Moving Vehicle Detection and Tracking System Based on Video Image," in *2013 Third International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC)*, 2013, pp. 1277–1280.
[72] C. Hou, H. Ai, and S. Lao, "Multiview pedestrian detection based on vector boosting," in *Computer Vision–ACCV 2007*, Springer, 2007, pp. 210–219.
[73] L. Mao, M. Xie, Y. Huang, and Y. Zhang, "Preceding vehicle detection using Histograms of Oriented Gradients," in *2010 International Conference on Communications, Circuits and Systems (ICCCAS)*, 2010, pp. 354–358.
[74] W. Hailuo, W. Bo, and L. Sun, "Pyramid Histogram of Oriented Gradient and Particles Swarm Optimization Based SVM for Vehicle Detection," in *2013 Seventh International Conference on Image and Graphics (ICIG)*, 2013, pp. 323–327.
[75] Z. Chen, K. Chen, and J. Chen, "Vehicle and Pedestrian Detection Using Support Vector Machine and Histogram of Oriented Gradients Features," in *2013 International Conference on Computer Sciences and Applications (CSA)*, 2013, pp. 365–368.
[76] P. E. Rybski, D. Huber, D. D. Morris, and R. Hoffman, "Visual classification of coarse vehicle orientation using Histogram of Oriented Gradients features," in *2010 IEEE Intelligent Vehicles Symposium (IV)*, 2010, pp. 921–928.
[77] T. Anakavej, A. Kawewong, and K. Patanukhom, "Internet-Vision Based Vehicle Model Query System Using Eigenfaces and Pyramid of Histogram of Oriented Gradients," in

2013 International Conference on Signal-Image Technology Internet-Based Systems (SITIS), 2013, pp. 179–186.
[78] Q. Tian, T. Zhong, and H. Li, "A new method for vehicle detection using MexicanHat wavelet and moment invariants," in *2013 IEEE Workshop on Signal Processing Systems (SiPS)*, 2013, pp. 289–294.
[79] C. Zhang, X. Chen, and W.-B. Chen, "A PCA-Based Vehicle Classification Framework," in *22nd International Conference on Data Engineering Workshops, 2006. Proceedings*, 2006, pp. 17–17.
[80] X. Mei, S. K. Zhou, and H. Wu, "Integrated Detection, Tracking and Recognition for IR Video-Based Vehicle Classification," in *2006 IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings*, 2006, vol. 5, pp. V–V.
[81] H. Choi and R. G. Baraniuk, "Multiscale image segmentation using wavelet-domain hidden Markov models," *Image Process. IEEE Trans. On*, vol. 10, no. 9, pp. 1309–1321, 2001.
[82] L. Walchshausl and R. Lindl, "Multi-Sensor Classification using a boosted Cascade Detector," in *2007 IEEE Intelligent Vehicles Symposium*, 2007, pp. 1045–1049.
[83] J. Li, W. Gong, W. Li, and X. Liu, "Robust pedestrian detection in thermal infrared imagery using the wavelet transform," *Infrared Phys. Technol.*, vol. 53, no. 4, pp. 267–273, 2010.
[84] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.
[85] R. DiSalle, "Space and Time: Inertial Frames," in *The Stanford Encyclopedia of Philosophy*, Winter 2009., E. N. Zalta, Ed. 2009.
[86] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, pp. 15–16, 2006.
[87] S. W. Smith, *The Scientist & Engineer's Guide to Digital Signal Processing*, 1st edition. San Diego, Calif: California Technical Pub, 1997.
[88] R. C. G. Chehata, W. B. Mikhael, and G. Atia, "A Transform Domain Modular approach for facial recognition using different representations and windowing techniques," in *2014 IEEE 57th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2014, pp. 817–820.
[89] R. C. G. Chehata, W. B. Mikhael, and M. M. Abdelwahab, "Performance evaluation of Transform Domain Diagonal Principal Component Analysis for facial recognition employing different pre-processing spatial domain approaches," in *2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2012, pp. 666–669.
[90] W. Mikhael and M. Abdelwahab, "A novel intelligent classifier/recognizer employing vector quantization coding of nonorthogonal signal and preprocessed signal representations," in *Proceedings of the 2003 International Conference on Neural Networks and Signal Processing, 2003*, 2003, vol. 1, pp. 826–829 Vol.1.
[91] W. B. Mikhael and P. Premakanthan, "Speaker recognition employing waveform based signal representation in nonorthogonal multiple transform domains," in *IEEE International Symposium on Circuits and Systems, 2002. ISCAS 2002*, 2002, vol. 2, pp. II–608–II–611 vol.2.
[92] R. O. Duda, *Pattern Classification 2nd Edition with Computer Manual 2nd Edition Set*, 2 edition. Wiley - 2006: Wiley-Interscience, 2004.

[93] J. Shlens, "A tutorial on principal component analysis," *ArXiv Prepr. ArXiv14041100*, 2014.
[94] W. K. McDowell, W. B. Mikhael, and A. P. Berg, "Efficiency of the KLT on Voiced amp; Unvoiced Speech as a function of segment size," in *2012 Proceedings of IEEE Southeastcon*, 2012, pp. 1–5.
[95] C. Egho and T. Vladimirova, "Adaptive hyperspectral image compression using the KLT and integer KLT algorithms," in *2014 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, 2014, pp. 112–119.
[96] P. Quintiliano and A. Santa-Rosa, "Detection of streets based on KLT using IKONOS multispectral images," in *2nd GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas, 2003*, 2003, pp. 186–190.
[97] X. Cao, J. Lan, P. Yan, and X. Li, "KLT Feature Based Vehicle Detection and Tracking in Airborne Videos," in *2011 Sixth International Conference on Image and Graphics (ICIG)*, 2011, pp. 673–678.
[98] R. Gutierrez-Osuna, "Lecture notes - Perception, Sensing & Instrumentation lab - Fisher Linear Discriminants," *Perception Sensing Instrumentation*. .
[99] T. Xiong and V. Cherkassky, "A combined SVM and LDA approach for classification," in *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on*, 2005, vol. 3, pp. 1455–1459.
[100] A. Subasi and M. I. Gursoy, "EEG signal classification using PCA, ICA, LDA and support vector machines," *Expert Syst. Appl.*, vol. 37, no. 12, pp. 8659–8666, 2010.
[101] J. Peng, D. R. Heisterkamp, and H. K. Dai, "LDA/SVM driven nearest neighbor classification," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 2001, vol. 1, pp. I–58.
[102] B. T. Morris and M. M. Trivedi, "Learning, modeling, and classification of vehicle track patterns from live video," *Intell. Transp. Syst. IEEE Trans. On*, vol. 9, no. 3, pp. 425–437, 2008.
[103] M. Greenacre, "Measures of distance between samples: Euclidean," *Corresp. Anal. Relat. Methods Stanf. Univ.*, 2008.
[104] G. Sun, C. Zhang, W. Zou, and G. Yu, "A new recognition method of vehicle license plate based on Genetic Neural Network," in *2010 the 5th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2010, pp. 1662–1666.
[105] L. Yu, L. Yu, J. Wang, L. Yu, Y. Qi, and H. Wen, "Back-Propagation Neural Network for Traffic Incident Detection Based on Fusion of Loop Detector and Probe Vehicle Data," in *Fourth International Conference on Natural Computation, 2008. ICNC '08*, 2008, vol. 3, pp. 116–120.
[106] L. Zhao and C. E. Thorpe, "Stereo-and neural network-based pedestrian detection," *Intell. Transp. Syst. IEEE Trans. On*, vol. 1, no. 3, pp. 148–154, 2000.
[107] W. Zhang, G. Tan, N. Ding, Y. Shang, and M. Lin, "Vehicle Classification Algorithm based on Binary Proximity Magnetic Sensors and Neural Network," in *11th International IEEE Conference on Intelligent Transportation Systems, 2008. ITSC 2008*, 2008, pp. 145–150.
[108] D. M. Ha, J.-M. Lee, and Y.-D. Kim, "Neural-edge-based vehicle detection and traffic parameter extraction," *Image Vis. Comput.*, vol. 22, no. 11, pp. 899–907, Sep. 2004.

[109] H. M. Ebeid, "Using MLP and RBF neural networks for face recognition: An insightful comparative case study," in *2011 International Conference on Computer Engineering Systems (ICCES)*, 2011, pp. 123–128.
[110] "*A priori* and *a posteriori*," *Wikipedia, the free encyclopedia*. 28-Apr-2015.
[111] Y. Zhang and Q. Ji, "Active and dynamic information fusion for multisensor systems with dynamic bayesian networks," *Syst. Man Cybern. Part B Cybern. IEEE Trans. On*, vol. 36, no. 2, pp. 467–472, 2006.
[112] E. F. Nakamura, A. A. Loureiro, and A. C. Frery, "Information fusion for wireless sensor networks: Methods, models, and classifications," *ACM Comput. Surv. CSUR*, vol. 39, no. 3, p. 9, 2007.
[113] N. Xiong and P. Svensson, "Multi-sensor management for information fusion: issues and approaches," *Inf. Fusion*, vol. 3, no. 2, pp. 163–186, 2002.
[114] I. Hara, F. Asano, H. Asoh, J. Ogata, N. Ichimura, Y. Kawai, F. Kanehiro, H. Hirukawa, and K. Yamamoto, "Robust speech interface based on audio and video information fusion for humanoid HRP-2," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, 2004, vol. 3, pp. 2404–2410.
[115] P. Helman, R. Veroff, S. R. Atlas, and C. Willman, "A Bayesian network classification methodology for gene expression data," *J. Comput. Biol.*, vol. 11, no. 4, pp. 581–615, 2004.
[116] A. McCallum, K. Nigam, and others, "A comparison of event models for naive bayes text classification," in *AAAI-98 workshop on learning for text categorization*, 1998, vol. 752, pp. 41–48.
[117] N. Friedman, M. Goldszmidt, and T. J. Lee, "Bayesian Network Classification with Continuous Attributes: Getting the Best of Both Discretization and Parametric Fitting.," in *ICML*, 1998, vol. 98, pp. 179–187.
[118] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Mach. Learn.*, vol. 29, no. 2–3, pp. 131–163, 1997.
[119] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," in *ACM SIGMETRICS Performance Evaluation Review*, 2005, vol. 33, pp. 50–60.
[120] K. P. Murphy, "Dynamic bayesian networks: representation, inference and learning," University of California, Berkeley, 2002.
[121] R. O. Duda, *Pattern Classification 2nd Edition with Computer Manual 2nd Edition Set*, 2 edition. Wiley - 2006: Wiley-Interscience, 2004.
[122] K. P. Murphy, "Bayes net toolbox for Matlab," in *Operations Research*, 1999.
[123] L. Mendelsohn, "MATLAB Neural Network Toolbox," *Prod. Rev. Tech. Anal. STOCKS Commod.*, vol. 12, 1994.
[124] P. Zarchan and H. Musoff, *Fundamentals of Kalman Filtering:: A Practical Approach*. Reston, Va: AIAA, 2009.
[125] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. John Wiley & Sons, 2004.
[126] "Kalman filter," *Wikipedia, the free encyclopedia*. 23-Apr-2015.
[127] A. Toloei and S. Niazi, "State Estimation for Target Tracking Problems with Nonlinear Kalman Filter Algorithms," *Int. J. Comput. Appl.*, vol. 98, no. 17, pp. 30–36, 2014.

[128] D. Kastaniotis, I. Theodorakopoulos, G. Economou, and S. Fotopoulos, "Gait-based gender recognition using pose information for real time applications," in *2013 18th International Conference on Digital Signal Processing (DSP)*, 2013, pp. 1–6.
[129] S. Chandran and A. Sawa, "Real-Time Detection and Understanding of Isolated Protruded Fingers," in *Conference on Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04*, 2004, pp. 152–152.
[130] J.-C. Lementec and P. Bajcsy, "Recognition of arm gestures using multiple orientation sensors: gesture classification," in *The 7th International IEEE Conference on Intelligent Transportation Systems, 2004. Proceedings*, 2004, pp. 965–970.